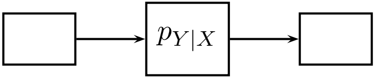
Towards an Algebraic Network Information Theory

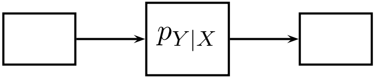
Bobak Nazer (BU)

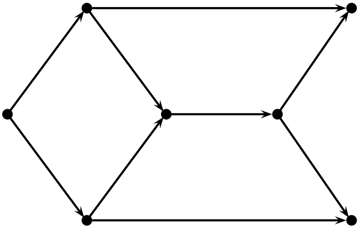
Tufts ECE Seminar
October 28, 2016

Motivation

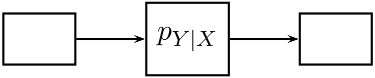


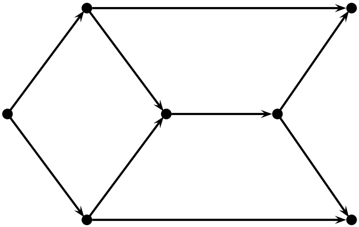
Motivation

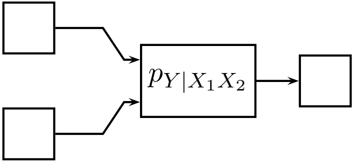




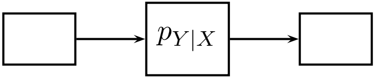
Motivation

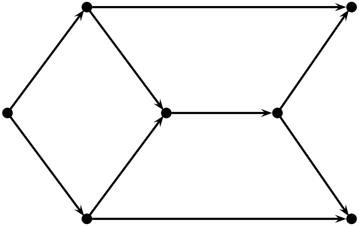


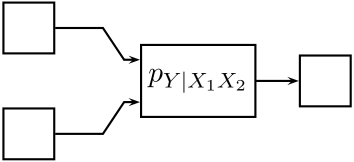


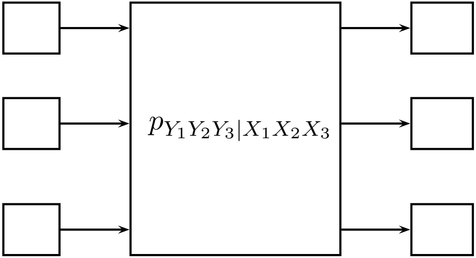


Motivation









Goal: Roughly speaking, for a given network, determine necessary and sufficient conditions on the rates at which the sources (or some functions thereof) can be communicated to the destinations.

Goal: Roughly speaking, for a given network, determine necessary and sufficient conditions on the rates at which the sources (or some functions thereof) can be communicated to the destinations.

Classical Approach:

- Use average performance of random i.i.d. codebooks to argue good codebooks exist.

Goal: Roughly speaking, for a given network, determine necessary and sufficient conditions on the rates at which the sources (or some functions thereof) can be communicated to the destinations.

Classical Approach:

- Use average performance of random i.i.d. codebooks to argue good codebooks exist.
- Powerful generalizations including superposition coding, dirty paper coding, block Markov coding, and many more...

Goal: Roughly speaking, for a given network, determine necessary and sufficient conditions on the rates at which the sources (or some functions thereof) can be communicated to the destinations.

Classical Approach:

- Use average performance of random i.i.d. codebooks to argue good codebooks exist.
- Powerful generalizations including superposition coding, dirty paper coding, block Markov coding, and many more...
- Rate regions described in terms of (single-letter) information measures optimized over pmfs.

Goal: Roughly speaking, for a given network, determine necessary and sufficient conditions on the rates at which the sources (or some functions thereof) can be communicated to the destinations.

Classical Approach:

- Use average performance of random i.i.d. codebooks to argue good codebooks exist.
- Powerful generalizations including superposition coding, dirty paper coding, block Markov coding, and many more...
- Rate regions described in terms of (single-letter) information measures optimized over pmfs.
- Many important successes: multiple-access channels, (degraded) broadcast channels, Slepian-Wolf compression, network coding, and many more...

Goal: Roughly speaking, for a given network, determine necessary and sufficient conditions on the rates at which the sources (or some functions thereof) can be communicated to the destinations.

Classical Approach:

- Use average performance of random i.i.d. codebooks to argue good codebooks exist.
- Powerful generalizations including superposition coding, dirty paper coding, block Markov coding, and many more...
- Rate regions described in terms of (single-letter) information measures optimized over pmfs.
- Many important successes: multiple-access channels, (degraded) broadcast channels, Slepian-Wolf compression, network coding, and many more...
- Guided the development and optimization of modern communication networks.

Goal: Roughly speaking, for a given network, determine necessary and sufficient conditions on the rates at which the sources (or some functions thereof) can be communicated to the destinations.

Classical Approach:

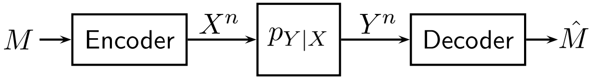
- Use average performance of random i.i.d. codebooks to argue good codebooks exist.
- Powerful generalizations including superposition coding, dirty paper coding, block Markov coding, and many more...
- Rate regions described in terms of (single-letter) information measures optimized over pmfs.
- Many important successes: multiple-access channels, (degraded) broadcast channels, Slepian-Wolf compression, network coding, and many more...
- Guided the development and optimization of modern communication networks.
- State-of-the-art elegantly captured in the recent textbook of **El Gamal and Kim.**

Goal: Roughly speaking, for a given network, determine necessary and sufficient conditions on the rates at which the sources (or some functions thereof) can be communicated to the destinations.

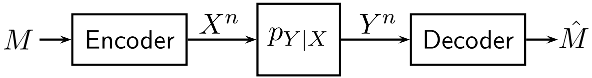
Classical Approach:

- Use average performance of random i.i.d. codebooks to argue good codebooks exist.
- Powerful generalizations including superposition coding, dirty paper coding, block Markov coding, and many more...
- Rate regions described in terms of (single-letter) information measures optimized over pmfs.
- Many important successes: multiple-access channels, (degraded) broadcast channels, Slepian-Wolf compression, network coding, and many more...
- Guided the development and optimization of modern communication networks.
- State-of-the-art elegantly captured in the recent textbook of **El Gamal and Kim.**
- Codes with algebraic structure are sought after to mimic the performance of random i.i.d. codes with low implementation complexity.

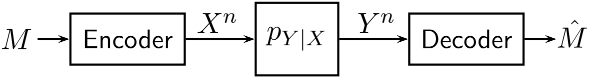
Point-to-Point Channels



Point-to-Point Channels

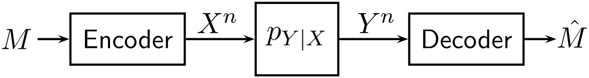


- Messages: $m \in [2^{nR}] \triangleq \{0, \ldots, 2^{nR} - 1\}$

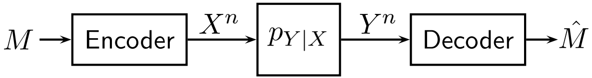


- Messages: $m \in [2^{nR}] \triangleq \{0, \ldots, 2^{nR} - 1\}$
- Encoder: a mapping $x^n(m) \in \mathcal{X}^n$ for each $m \in [2^{nR}]$

Point-to-Point Channels

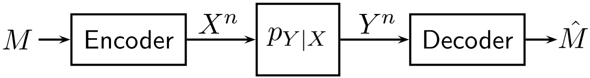


- Messages: $m \in [2^{nR}] \triangleq \{0, \ldots, 2^{nR} - 1\}$
- Encoder: a mapping $x^n(m) \in \mathcal{X}^n$ for each $m \in [2^{nR}]$
- Memoryless Channel: $p_{Y^n|X^n}(y^n|x^n) = \prod_{i=1}^{n} p_{Y|X}(y_i|x_i)$



- Messages: $m \in [2^{nR}] \triangleq \{0, \ldots, 2^{nR} - 1\}$
- Encoder: a mapping $x^n(m) \in \mathcal{X}^n$ for each $m \in [2^{nR}]$
- Memoryless Channel: $p_{Y^n|X^n}(y^n|x^n) = \prod_{i=1}^{n} p_{Y|X}(y_i|x_i)$
- Decoder: a mapping $\hat{m}(y^n) \in [2^{nR}]$ for each $y^n \in \mathcal{Y}^n$

Point-to-Point Channels

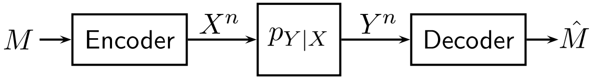


- Messages: $m \in [2^{nR}] \triangleq \{0, \ldots, 2^{nR} - 1\}$
- Encoder: a mapping $x^n(m) \in \mathcal{X}^n$ for each $m \in [2^{nR}]$
- Memoryless Channel: $p_{Y^n|X^n}(y^n|x^n) = \prod_{i=1}^{n} p_{Y|X}(y_i|x_i)$
- Decoder: a mapping $\hat{m}(y^n) \in [2^{nR}]$ for each $y^n \in \mathcal{Y}^n$

Theorem (Shannon '48)

$$C = \max_{p_X(x)} I(X;Y)$$

Point-to-Point Channels



- Messages: $m \in [2^{nR}] \triangleq \{0, \ldots, 2^{nR} - 1\}$
- Encoder: a mapping $x^n(m) \in \mathcal{X}^n$ for each $m \in [2^{nR}]$
- Memoryless Channel: $p_{Y^n|X^n}(y^n|x^n) = \prod_{i=1}^{n} p_{Y|X}(y_i|x_i)$
- Decoder: a mapping $\hat{m}(y^n) \in [2^{nR}]$ for each $y^n \in \mathcal{Y}^n$

Theorem (Shannon '48)

$$C = \max_{p_X(x)} I(X;Y)$$

- Proof relies on random i.i.d. codebooks combined with joint typicality decoding.

Typicality

- x^n is a length-n sequence with elements from finite alphabet $\mathcal{X}$
- The empirical pmf (i.e., type) of x^n is

$$\pi(x|x^n) = \frac{1}{n}\big|\{i : x_i = x\}\big| \quad x \in \mathcal{X}$$

- x^n is a length-n sequence with elements from finite alphabet $\mathcal{X}$
- The empirical pmf (i.e., type) of x^n is

$$\pi(x|x^n) = \frac{1}{n}\big|\{i : x_i = x\}\big| \quad x \in \mathcal{X}$$

- If X^n is i.i.d. according to $p_X(x)$, then, by the weak law of large numbers, $\pi(x|x^n)$ converges to $p_X(x)$ in probability.

Typicality

- x^n is a length-n sequence with elements from finite alphabet $\mathcal{X}$
- The empirical pmf (i.e., type) of x^n is

$$\pi(x|x^n) = \frac{1}{n}\big|\{i : x_i = x\}\big| \quad x \in \mathcal{X}$$

- If X^n is i.i.d. according to $p_X(x)$, then, by the weak law of large numbers, $\pi(x|x^n)$ converges to $p_X(x)$ in probability.
- This motivates the typical set

$$\mathcal{T}_\epsilon^{(n)}(X) = \big\{x^n : |\pi(x|x^n) - p_X(x)| \le \epsilon p_X(x) \;\text{ for all } x \in \mathcal{X}\big\}$$

which satisfies $\lim_{n\to\infty} \mathsf{P}\big(X^n \in \mathcal{T}_\epsilon^{(n)}\big) = 1$.

Typicality

- x^n is a length-n sequence with elements from finite alphabet $\mathcal{X}$
- The empirical pmf (i.e., type) of x^n is

$$\pi(x|x^n) = \frac{1}{n}\big|\{i : x_i = x\}\big| \quad x \in \mathcal{X}$$

- If X^n is i.i.d. according to $p_X(x)$, then, by the weak law of large numbers, $\pi(x|x^n)$ converges to $p_X(x)$ in probability.
- This motivates the typical set

$$\mathcal{T}_\epsilon^{(n)}(X) = \big\{x^n : |\pi(x|x^n) - p_X(x)| \le \epsilon p_X(x) \ \text{ for all } x \in \mathcal{X}\big\}$$

  which satisfies $\lim_{n\to\infty} \mathsf{P}\big(X^n \in \mathcal{T}_\epsilon^{(n)}\big) = 1$.
- We can generalize this definition to pairs of sequences (X^n, Y^n) that are i.i.d. according to $p_{XY}(x, y)$ and so on...

- Joint typicality is a powerful framework due to the availability of several key lemmas including

- Joint typicality is a powerful framework due to the availability of several key lemmas including

Joint Typicality Lemma

Select $p_{XY}(x, y)$ and $0 < \epsilon' < \epsilon$. Then, there exists $\delta(\epsilon)$ that tends to 0 as $\epsilon \to 0$ such that

- Joint typicality is a powerful framework due to the availability of several key lemmas including

Joint Typicality Lemma

Select $p_{XY}(x,y)$ and $0 < \epsilon' < \epsilon$. Then, there exists $\delta(\epsilon)$ that tends to 0 as $\epsilon \to 0$ such that

- For any $\tilde{y}^n \in \mathcal{Y}^n$ and $\tilde{X}^n$ i.i.d. $p_X(\tilde{x})$,

$$\mathsf{P}\big\{(\tilde{X}^n, \tilde{y}^n) \in \mathcal{T}_\epsilon^{(n)}(X,Y)\big\} \le 2^{-n(I(X;Y)-\delta(\epsilon))}$$

Joint Typicality Lemma

- Joint typicality is a powerful framework due to the availability of several key lemmas including

Joint Typicality Lemma

Select $p_{XY}(x, y)$ and $0 < \epsilon' < \epsilon$. Then, there exists $\delta(\epsilon)$ that tends to 0 as $\epsilon \to 0$ such that

- For any $\tilde{y}^n \in \mathcal{Y}^n$ and $\tilde{X}^n$ i.i.d. $p_X(\tilde{x})$,

$$\mathsf{P}\{(\tilde{X}^n, \tilde{y}^n) \in \mathcal{T}_\epsilon^{(n)}(X, Y)\} \le 2^{-n(I(X;Y)-\delta(\epsilon))}$$

- For any $y^n \in \mathcal{T}_{\epsilon'}^{(n)}(Y)$ and $\tilde{X}^n$ i.i.d. $p_X(\tilde{x})$,

$$\mathsf{P}\{(\tilde{X}^n, y^n) \in \mathcal{T}_\epsilon^{(n)}(X, Y)\} \ge 2^{-n(I(X;Y)+\delta(\epsilon))} \ .$$

Joint Typicality Lemma

- Joint typicality is a powerful framework due to the availability of several key lemmas including

Joint Typicality Lemma

Select $p_{XY}(x, y)$ and $0 < \epsilon' < \epsilon$. Then, there exists $\delta(\epsilon)$ that tends to 0 as $\epsilon \to 0$ such that

- For any $\tilde{y}^n \in \mathcal{Y}^n$ and $\tilde{X}^n$ i.i.d. $p_X(\tilde{x})$,

$$\mathsf{P}\big\{(\tilde{X}^n, \tilde{y}^n) \in \mathcal{T}_\epsilon^{(n)}(X, Y)\big\} \le 2^{-n(I(X;Y)-\delta(\epsilon))}$$

- For any $y^n \in \mathcal{T}_{\epsilon'}^{(n)}(Y)$ and $\tilde{X}^n$ i.i.d. $p_X(\tilde{x})$,

$$\mathsf{P}\big\{(\tilde{X}^n, y^n) \in \mathcal{T}_\epsilon^{(n)}(X, Y)\big\} \ge 2^{-n(I(X;Y)+\delta(\epsilon))} \; .$$

Intuition: Probability that i.i.d. $\tilde{X}^n$ looks jointly typical $\approx 2^{-nI(X;Y)}$

Point-to-Point Capacity: Achievability Proof

- **Code Construction:** Generate 2^{nR} random codewords $X^n(1), \ldots, X^n(2^{nR})$ with each element drawn i.i.d. $p_X(x)$.

Point-to-Point Capacity: Achievability Proof

- **Code Construction:** Generate 2^{nR} random codewords $X^n(1), \ldots, X^n(2^{nR})$ with each element drawn i.i.d. $p_X(x)$.
- **Encoding:** For message $m \in [2^{nR}]$, send codeword $X^n(m)$.

Point-to-Point Capacity: Achievability Proof

- **Code Construction:** Generate 2^{nR} random codewords $X^n(1), \ldots, X^n(2^{nR})$ with each element drawn i.i.d. $p_X(x)$.
- **Encoding:** For message $m \in [2^{nR}]$, send codeword $X^n(m)$.
- **Decoding:** Search for $\hat{m}$ such that $(X^n(\hat{m}), Y^n)$ is jointly typical. If only one such $\hat{m}$, output it as the message estimate. Otherwise, declare an error.

Point-to-Point Capacity: Achievability Proof

- **Code Construction:** Generate 2^{nR} random codewords $X^n(1), \ldots, X^n(2^{nR})$ with each element drawn i.i.d. $p_X(x)$.
- **Encoding:** For message $m \in [2^{nR}]$, send codeword $X^n(m)$.
- **Decoding:** Search for $\hat{m}$ such that $(X^n(\hat{m}), Y^n)$ is jointly typical. If only one such $\hat{m}$, output it as the message estimate. Otherwise, declare an error.
- **Error Analysis:** Two possibilities.

Point-to-Point Capacity: Achievability Proof

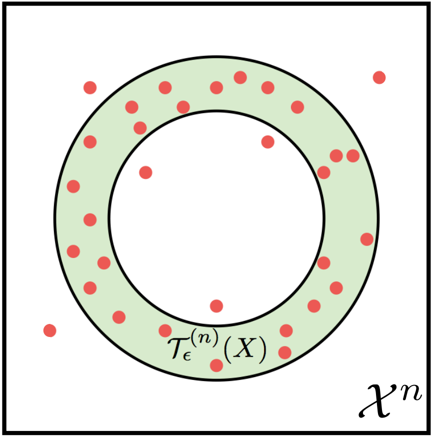
- **Code Construction:** Generate 2^{nR} random codewords $X^n(1), \ldots, X^n(2^{nR})$ with each element drawn i.i.d. $p_X(x)$.
- **Encoding:** For message $m \in [2^{nR}]$, send codeword $X^n(m)$.
- **Decoding:** Search for $\hat{m}$ such that $(X^n(\hat{m}), Y^n)$ is jointly typical. If only one such $\hat{m}$, output it as the message estimate. Otherwise, declare an error.
- **Error Analysis:** Two possibilities.
  - True codeword is not jointly typical, $(X^n(m), Y^n) \notin \mathcal{T}_\epsilon^{(n)}$. Probability goes to zero via WLLN.

Point-to-Point Capacity: Achievability Proof

- **Code Construction:** Generate 2^{nR} random codewords $X^n(1), \ldots, X^n(2^{nR})$ with each element drawn i.i.d. $p_X(x)$.
- **Encoding:** For message $m \in [2^{nR}]$, send codeword $X^n(m)$.
- **Decoding:** Search for $\hat{m}$ such that $(X^n(\hat{m}), Y^n)$ is jointly typical. If only one such $\hat{m}$, output it as the message estimate. Otherwise, declare an error.
- **Error Analysis:** Two possibilities.
  - True codeword is not jointly typical, $(X^n(m), Y^n) \notin \mathcal{T}_\epsilon^{(n)}$. Probability goes to zero via WLLN.
  - Some other codeword is jointly typical,

$$\begin{aligned}\mathsf{P}\Big\{ \bigcup_{\tilde{m} \neq m} \big\{(X^n(\tilde{m}), Y^n) \in \mathcal{T}_\epsilon^{(n)}\big\}\Big\} &\leq \sum_{\tilde{m} \neq m} \mathsf{P}\big\{(X^n(\tilde{m}), Y^n) \in \mathcal{T}_\epsilon^{(n)}\big\} \\ &\leq \sum_{\tilde{m} \neq m} 2^{-nI(X;Y) - \delta(\epsilon)} \\ &< 2^{nR}\, 2^{-nI(X;Y) - \delta(\epsilon)}\ .\end{aligned}$$

    Probability goes to zero if $R < I(X;Y) - \delta(\epsilon)$.



Random i.i.d. Codes

- Codewords are independent of one another.
- Can directly target an input distribution $p_X(x)$.

Goal: Roughly speaking, for a given network, determine necessary and sufficient conditions on the rates at which the sources (or some functions thereof) can be communicated to the destinations.

Goal: Roughly speaking, for a given network, determine necessary and sufficient conditions on the rates at which the sources (or some functions thereof) can be communicated to the destinations.

Algebraic Approach:

- Utilize linear or lattice codebooks.

Goal: Roughly speaking, for a given network, determine necessary and sufficient conditions on the rates at which the sources (or some functions thereof) can be communicated to the destinations.

Algebraic Approach:

- Utilize linear or lattice codebooks.
- Compelling examples starting from the work of Körner and Marton on distributed compression and, more recently, many papers on physical-layer network coding, distributed dirty paper coding, and interference alignment.

Goal: Roughly speaking, for a given network, determine necessary and sufficient conditions on the rates at which the sources (or some functions thereof) can be communicated to the destinations.

Algebraic Approach:

- Utilize linear or lattice codebooks.
- Compelling examples starting from the work of Körner and Marton on distributed compression and, more recently, many papers on physical-layer network coding, distributed dirty paper coding, and interference alignment.
- Coding schemes exhibit behavior not found via i.i.d. ensembles.

Goal: Roughly speaking, for a given network, determine necessary and sufficient conditions on the rates at which the sources (or some functions thereof) can be communicated to the destinations.

Algebraic Approach:

- Utilize linear or lattice codebooks.
- Compelling examples starting from the work of Körner and Marton on distributed compression and, more recently, many papers on physical-layer network coding, distributed dirty paper coding, and interference alignment.
- Coding schemes exhibit behavior not found via i.i.d. ensembles.
- However, some classical coding techniques are still unavailable.

Goal: Roughly speaking, for a given network, determine necessary and sufficient conditions on the rates at which the sources (or some functions thereof) can be communicated to the destinations.

Algebraic Approach:

- Utilize linear or lattice codebooks.
- Compelling examples starting from the work of Körner and Marton on distributed compression and, more recently, many papers on physical-layer network coding, distributed dirty paper coding, and interference alignment.
- Coding schemes exhibit behavior not found via i.i.d. ensembles.
- However, some classical coding techniques are still unavailable.
- Most of the initial efforts have focused on Gaussian networks and have employed nested lattice codebooks.

Goal: Roughly speaking, for a given network, determine necessary and sufficient conditions on the rates at which the sources (or some functions thereof) can be communicated to the destinations.

Algebraic Approach:

- Utilize linear or lattice codebooks.
- Compelling examples starting from the work of Körner and Marton on distributed compression and, more recently, many papers on physical-layer network coding, distributed dirty paper coding, and interference alignment.
- Coding schemes exhibit behavior not found via i.i.d. ensembles.
- However, some classical coding techniques are still unavailable.
- Most of the initial efforts have focused on Gaussian networks and have employed nested lattice codebooks.
- Are these just a collection of intriguing examples or elements of a more general theory?

Goal: Roughly speaking, for a given network, determine necessary and sufficient conditions on the rates at which the sources (or some functions thereof) can be communicated to the destinations.

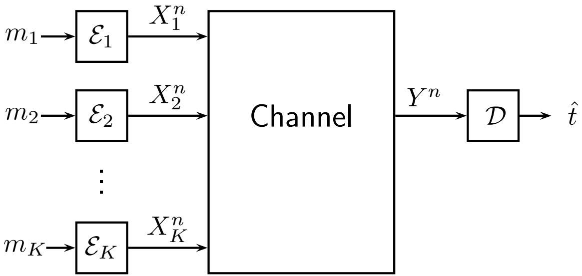
Algebraic Approach:

- Utilize linear or lattice codebooks.
- Compelling examples starting from the work of Körner and Marton on distributed compression and, more recently, many papers on physical-layer network coding, distributed dirty paper coding, and interference alignment.
- Coding schemes exhibit behavior not found via i.i.d. ensembles.
- However, some classical coding techniques are still unavailable.
- Most of the initial efforts have focused on Gaussian networks and have employed nested lattice codebooks.
- Are these just a collection of intriguing examples or elements of a more general theory?

This Talk: We build on previous work and propose a joint typicality approach to algebraic network information theory.

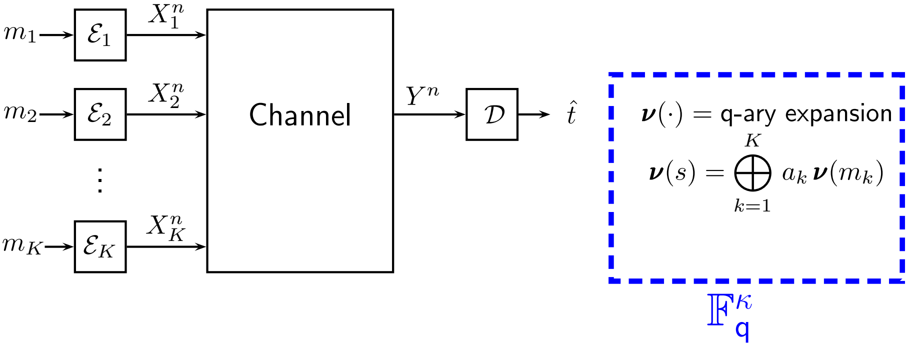
Goal: Send a linear combination of the messages to the receiver.

Goal: Send a linear combination of the messages to the receiver.

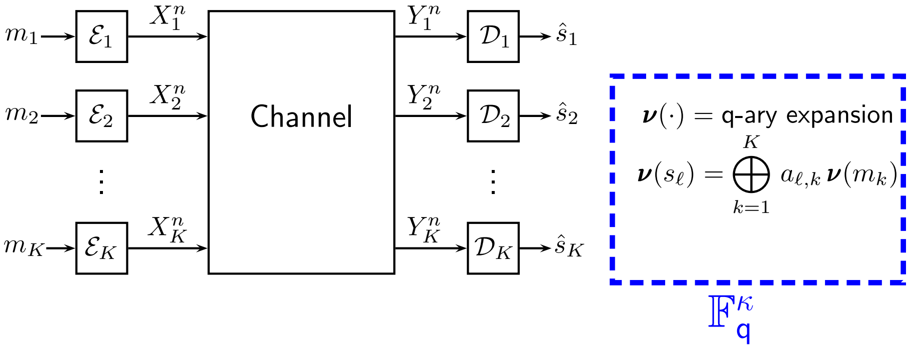


Compute-and-Forward

Goal: Send a linear combination of the messages to the receiver.



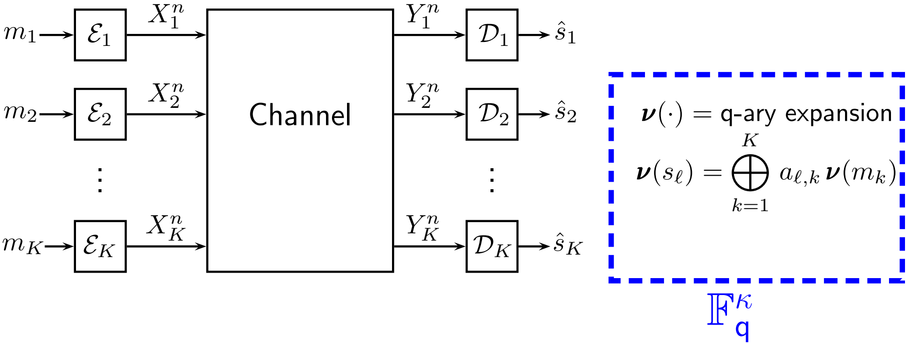
Goal: Send linear combinations of the messages to the receivers.



Compute-and-Forward

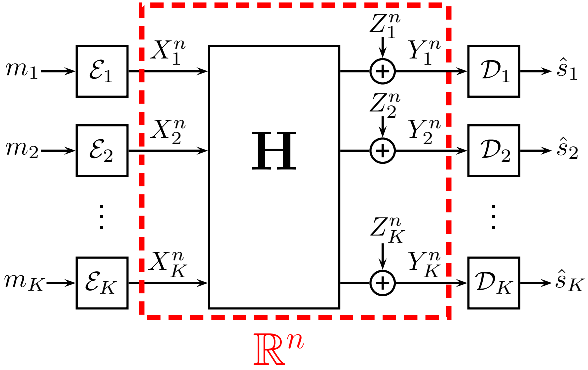
Goal: Send linear combinations of the messages to the receivers.

- Compute-and-forward can serve as a framework for communicating messages across a network (e.g., relaying, MIMO uplink/downlink, interference alignment).



Goal: Send linear combinations of the messages to the receivers.

- Compute-and-forward can serve as a framework for communicating messages across a network (e.g., relaying, MIMO uplink/downlink, interference alignment).
- Much of the recent work has focused on Gaussian networks.



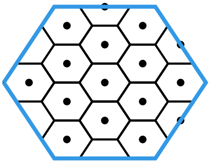
$\boldsymbol{\nu}(\cdot)$ = q-ary expansion

$$\boldsymbol{\nu}(s_\ell) = \bigoplus_{k=1}^{K} a_{\ell,k}\, \boldsymbol{\nu}(m_k)$$

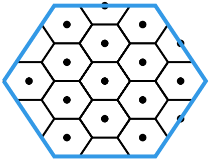
$\mathbb{F}_{\mathsf{q}}^{\kappa}$

The Usual Approach

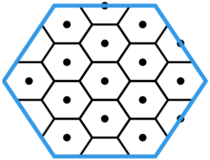
- Nested Lattice Code: Formed by taking all elements of Λ_{F} that lie in the fundamental Voronoi region of Λ_{C}.



- Nested Lattice Code: Formed by taking all elements of Λ_{F} that lie in the fundamental Voronoi region of Λ_{C}.
- Fine lattice Λ_{F} protects against noise.

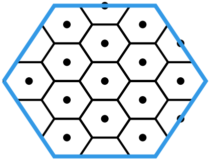


- Nested Lattice Code: Formed by taking all elements of Λ_{F} that lie in the fundamental Voronoi region of Λ_{C}.
- Fine lattice Λ_{F} protects against noise.
- Coarse lattice Λ_{C} enforces the power constraint.



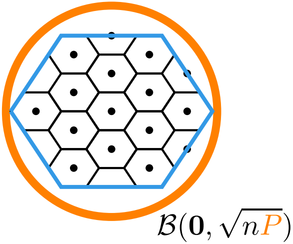
Nested Lattice Codes

- Nested Lattice Code: Formed by taking all elements of Λ_F that lie in the fundamental Voronoi region of Λ_C.
- Fine lattice Λ_F protects against noise.
- Coarse lattice Λ_C enforces the power constraint.
- Existence of good nested lattice codes: **Loeliger '97, Forney-Trott-Chung '00, Erez-Litsyn-Zamir '05, Ordentlich-Erez '16.**
- **Erez-Zamir '04:** Nested lattice codes can achieve the Gaussian capacity.
- **Zamir-Shamai-Erez '02:** Excellent framework for multi-terminal binning.

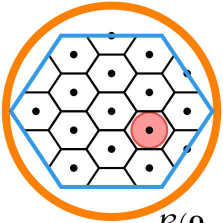


Nested Lattice Codes

- Nested Lattice Code: Formed by taking all elements of Λ_{F} that lie in the fundamental Voronoi region of Λ_{C}.
- Fine lattice Λ_{F} protects against noise.
- Coarse lattice Λ_{C} enforces the power constraint.
- Existence of good nested lattice codes: **Loeliger '97, Forney-Trott-Chung '00, Erez-Litsyn-Zamir '05, Ordentlich-Erez '16.**
- **Erez-Zamir '04:** Nested lattice codes can achieve the Gaussian capacity.
- **Zamir-Shamai-Erez '02:** Excellent framework for multi-terminal binning.



- Nested Lattice Code: Formed by taking all elements of Λ_{F} that lie in the fundamental Voronoi region of Λ_{C}.
- Fine lattice Λ_{F} protects against noise.
- Coarse lattice Λ_{C} enforces the power constraint.
- Existence of good nested lattice codes: **Loeliger '97, Forney-Trott-Chung '00, Erez-Litsyn-Zamir '05, Ordentlich-Erez '16.**
- **Erez-Zamir '04:** Nested lattice codes can achieve the Gaussian capacity.
- **Zamir-Shamai-Erez '02:** Excellent framework for multi-terminal binning.



Nested Lattice Code Heuristics

- The Voronoi region $\mathcal{V}_\mathsf{C}$ of the coarse lattice Λ_C enforces the power constraint: If $\mathbf{x} \in \mathcal{V}_\mathsf{C}$, then $\frac{1}{n}\|\mathbf{x}\|^2 \leq P$.

Nested Lattice Code Heuristics

- The Voronoi region $\mathcal{V}_{\mathsf{C}}$ of the coarse lattice Λ_{C} enforces the power constraint: If $\mathbf{x} \in \mathcal{V}_{\mathsf{C}}$, then $\frac{1}{n}\|\mathbf{x}\|^2 \leq P$.

- The Voronoi region $\mathcal{V}_{\mathsf{F}}$ of the fine lattice Λ_{F} tolerates noise up to variance σ^2_{eff}: For "well-behaved" noise $\mathbf{z}_{\text{eff}}$, if $\frac{1}{n}\mathbb{E}\|\mathbf{z}_{\text{eff}}\|^2 \leq \sigma^2_{\text{eff}}$, then $\mathsf{P}(\mathbf{z}_{\text{eff}} \notin \mathcal{V}_{\mathsf{F}}) < \delta$ for some small δ.

- The Voronoi region $\mathcal{V}_{\mathsf{C}}$ of the coarse lattice Λ_{C} enforces the power constraint: If $\mathbf{x} \in \mathcal{V}_{\mathsf{C}}$, then $\frac{1}{n}\|\mathbf{x}\|^2 \leq P$.

- The Voronoi region $\mathcal{V}_{\mathsf{F}}$ of the fine lattice Λ_{F} tolerates noise up to variance σ^2_{eff}: For "well-behaved" noise $\mathbf{z}_{\mathsf{eff}}$, if $\frac{1}{n}\mathbb{E}\|\mathbf{z}_{\mathsf{eff}}\|^2 \leq \sigma^2_{\mathsf{eff}}$, then $\mathsf{P}(\mathbf{z}_{\mathsf{eff}} \notin \mathcal{V}_{\mathsf{F}}) < \delta$ for some small δ.

- The number of codewords in the nested lattice codebook $\Lambda_{\mathsf{F}} \cap \mathcal{V}_{\mathsf{C}}$ is

$$2^{nR} = \frac{\mathrm{Vol}(\mathcal{V}_{\mathsf{C}})}{\mathrm{Vol}(\mathcal{V}_{\mathsf{F}})} \approx \frac{\mathrm{Vol}\Big(\mathcal{B}\big(\mathbf{0}, \sqrt{nP}\big)\Big)}{\mathrm{Vol}\Big(\mathcal{B}\big(\mathbf{0}, \sqrt{n\sigma^2_{\mathsf{eff}}}\big)\Big)} = \left(\frac{P}{\sigma^2_{\mathsf{eff}}}\right)^{n/2}$$

Nested Lattice Code Heuristics

- The Voronoi region $\mathcal{V}_{\mathsf{C}}$ of the coarse lattice Λ_{C} enforces the power constraint: If $\mathbf{x} \in \mathcal{V}_{\mathsf{C}}$, then $\frac{1}{n}\|\mathbf{x}\|^2 \leq P$.

- The Voronoi region $\mathcal{V}_{\mathsf{F}}$ of the fine lattice Λ_{F} tolerates noise up to variance σ^2_{eff}: For "well-behaved" noise $\mathbf{z}_{\text{eff}}$, if $\frac{1}{n}\mathbb{E}\|\mathbf{z}_{\text{eff}}\|^2 \leq \sigma^2_{\text{eff}}$, then $\mathsf{P}(\mathbf{z}_{\text{eff}} \notin \mathcal{V}_{\mathsf{F}}) < \delta$ for some small δ.

- The number of codewords in the nested lattice codebook $\Lambda_{\mathsf{F}} \cap \mathcal{V}_{\mathsf{C}}$ is

$$2^{nR} = \frac{\text{Vol}(\mathcal{V}_{\mathsf{C}})}{\text{Vol}(\mathcal{V}_{\mathsf{F}})} \approx \frac{\text{Vol}\Big(\mathcal{B}\big(\mathbf{0}, \sqrt{nP}\big)\Big)}{\text{Vol}\Big(\mathcal{B}\big(\mathbf{0}, \sqrt{n\sigma^2_{\text{eff}}}\big)\Big)} = \left(\frac{P}{\sigma^2_{\text{eff}}}\right)^{n/2}$$

- Can show that the achievable rate satisfies $R > \frac{1}{2}\log\left(\frac{P}{\sigma^2_{\text{eff}}}\right) - \delta$.

Compute-and-Forward with Lattice Codes

- Each encoder maps its message m_k to a lattice codeword $\mathbf{x}_k$.

- Each encoder maps its message m_k to a lattice codeword $\mathbf{x}_k$.
- The decoder observes $\mathbf{y}$. It scales by $\alpha \in \mathbb{R}$ to get $\mathbf{h} \in \mathbb{R}^K$ "closer" to $\boldsymbol{a} \in \mathbb{Z}^K$. We can write this as

$$\alpha \mathbf{y} =$$

- Each encoder maps its message m_k to a lattice codeword $\mathbf{x}_k$.
- The decoder observes $\mathbf{y}$. It scales by $\alpha \in \mathbb{R}$ to get $\mathbf{h} \in \mathbb{R}^K$ "closer" to $\boldsymbol{a} \in \mathbb{Z}^K$. We can write this as

$$\alpha \mathbf{y} = \underbrace{\sum_{k=1}^{K} a_k \mathbf{x}_k}_{\text{Lattice Codeword}} + \underbrace{\sum_{\ell=1}^{L} (\alpha h_k - a_k)\mathbf{x}_k + \alpha \mathbf{z}}_{\text{Effective Noise}}$$

- Each encoder maps its message m_k to a lattice codeword $\mathbf{x}_k$.
- The decoder observes $\mathbf{y}$. It scales by $\alpha \in \mathbb{R}$ to get $\mathbf{h} \in \mathbb{R}^K$ "closer" to $\boldsymbol{a} \in \mathbb{Z}^K$. We can write this as

$$\alpha \mathbf{y} = \underbrace{\sum_{k=1}^{K} a_k \mathbf{x}_k}_{\text{Lattice Codeword}} + \underbrace{\sum_{\ell=1}^{L} (\alpha h_k - a_k)\mathbf{x}_k + \alpha \mathbf{z}}_{\text{Effective Noise}} = \mathbf{v} + \mathbf{z}_{\text{eff}}$$

- Each encoder maps its message m_k to a lattice codeword $\mathbf{x}_k$.
- The decoder observes $\mathbf{y}$. It scales by $\alpha \in \mathbb{R}$ to get $\mathbf{h} \in \mathbb{R}^K$ "closer" to $\boldsymbol{a} \in \mathbb{Z}^K$. We can write this as

$$\alpha \mathbf{y} = \underbrace{\sum_{k=1}^{K} a_k \mathbf{x}_k}_{\text{Lattice Codeword}} + \underbrace{\sum_{\ell=1}^{L} (\alpha h_k - a_k)\mathbf{x}_k + \alpha \mathbf{z}}_{\text{Effective Noise}} = \mathbf{v} + \mathbf{z}_{\text{eff}}$$

- The effective noise variance is

$$\sigma^2_{\text{eff}} = \frac{1}{n}\mathbb{E}\|\mathbf{z}_{\text{eff}}\|^2 = \alpha^2 + \text{SNR} \sum_{k=1}^{K} (\alpha h_k - a_k)^2 = \alpha^2 + \text{SNR}\|\alpha \mathbf{h} - \boldsymbol{a}\|^2 .$$

Compute-and-Forward with Lattice Codes

- Each encoder maps its message m_k to a lattice codeword $\mathbf{x}_k$.
- The decoder observes $\mathbf{y}$. It scales by $\alpha \in \mathbb{R}$ to get $\mathbf{h} \in \mathbb{R}^K$ "closer" to $\boldsymbol{a} \in \mathbb{Z}^K$. We can write this as

$$\alpha \mathbf{y} = \underbrace{\sum_{k=1}^{K} a_k \mathbf{x}_k}_{\text{Lattice Codeword}} + \underbrace{\sum_{\ell=1}^{L} (\alpha h_k - a_k)\mathbf{x}_k + \alpha \mathbf{z}}_{\text{Effective Noise}} = \mathbf{v} + \mathbf{z}_{\text{eff}}$$

- The effective noise variance is

$$\sigma^2_{\text{eff}} = \frac{1}{n}\mathbb{E}\|\mathbf{z}_{\text{eff}}\|^2 = \alpha^2 + \text{SNR}\sum_{k=1}^{K}(\alpha h_k - a_k)^2 = \alpha^2 + \text{SNR}\|\alpha \mathbf{h} - \boldsymbol{a}\|^2 .$$

- We can decode $\mathbf{v}$ if $R < \frac{1}{2}\log\left(\frac{\text{SNR}}{\alpha^2 + \text{SNR}\|\alpha \mathbf{h} - \boldsymbol{a}\|^2}\right)$.

Compute-and-Forward with Lattice Codes

- Each encoder maps its message m_k to a lattice codeword $\mathbf{x}_k$.
- The decoder observes $\mathbf{y}$. It scales by $\alpha \in \mathbb{R}$ to get $\mathbf{h} \in \mathbb{R}^K$ "closer" to $\boldsymbol{a} \in \mathbb{Z}^K$. We can write this as

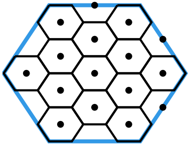
$$\alpha \mathbf{y} = \underbrace{\sum_{k=1}^{K} a_k \mathbf{x}_k}_{\text{Lattice Codeword}} + \underbrace{\sum_{\ell=1}^{L} (\alpha h_k - a_k)\mathbf{x}_k + \alpha \mathbf{z}}_{\text{Effective Noise}} = \mathbf{v} + \mathbf{z}_{\text{eff}}$$

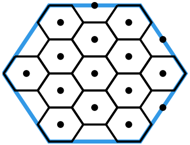
- The effective noise variance is

$$\sigma^2_{\text{eff}} = \frac{1}{n}\mathbb{E}\|\mathbf{z}_{\text{eff}}\|^2 = \alpha^2 + \text{SNR}\sum_{k=1}^{K}(\alpha h_k - a_k)^2 = \alpha^2 + \text{SNR}\|\alpha\mathbf{h} - \boldsymbol{a}\|^2 .$$

- We can decode $\mathbf{v}$ if $R < \frac{1}{2}\log\left(\frac{\text{SNR}}{\alpha^2 + \text{SNR}\|\alpha\mathbf{h} - \boldsymbol{a}\|^2}\right)$.
- Finding the best $\boldsymbol{a}$ corresponds to finding the shortest vector in the lattice $(\text{SNR}^{-1}\mathbf{I} + \mathbf{h}\mathbf{h}^{\mathsf{T}})^{-1/2}\mathbb{Z}^K$.

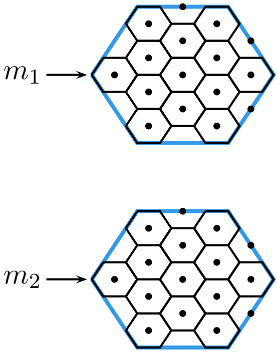
All users employ the same nested lattice code.





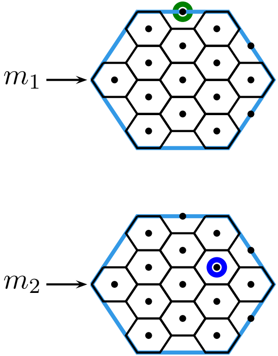
Compute-and-Forward: Illustration

Choose messages $m_k \in [2^{nR}]$.

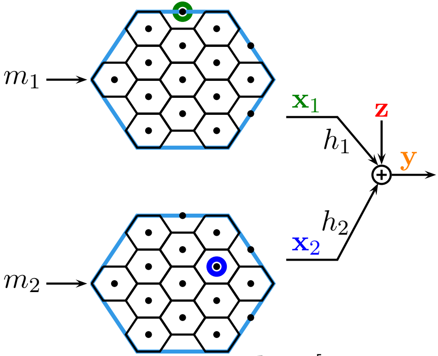


Compute-and-Forward: Illustration

Map m_k to lattice codeword $\mathbf{x}_k = \mathcal{E}_k(m_k)$.



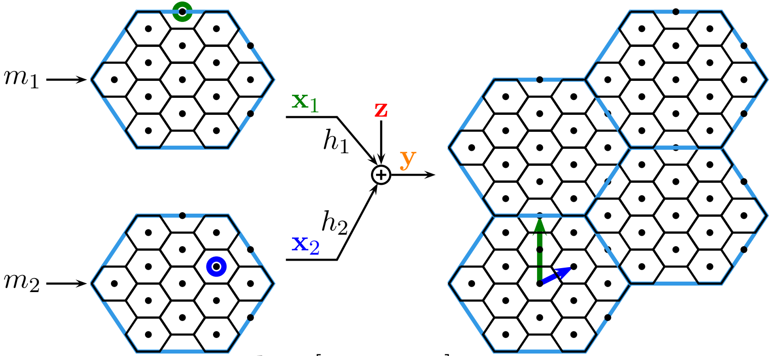
Transmit lattice points over the channel.



$$\mathbf{h} = [\ 1.4 \quad 2.1\ ]$$

$$\boldsymbol{a} = [\ 2 \quad 3\ ]$$

Transmit lattice points over the channel.

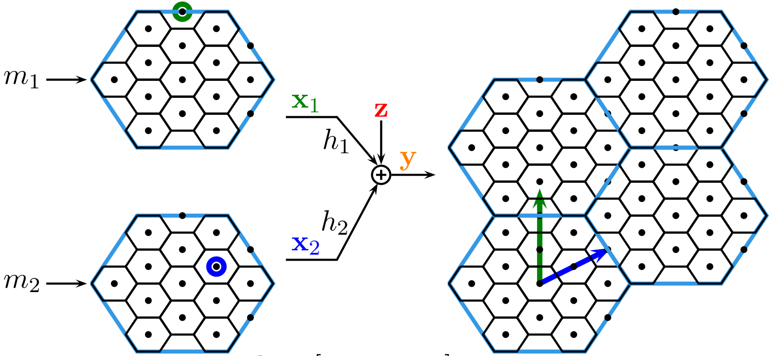


$$\mathbf{h} = [\ 1.4 \quad 2.1\ ]$$

$$\boldsymbol{a} = [\ 2 \quad 3\ ]$$

Compute-and-Forward: Illustration

Lattice codewords are scaled by channel coefficients.

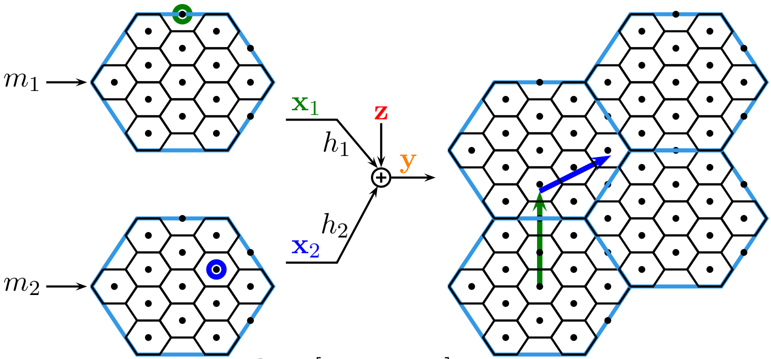


$$\mathbf{h} = [\ 1.4 \quad 2.1\ ]$$

$$\boldsymbol{a} = [\ 2 \quad 3\ ]$$

Compute-and-Forward: Illustration

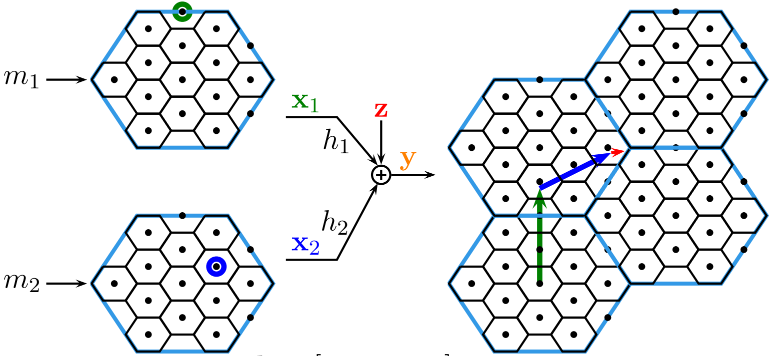
Scaled codewords added together plus noise.



$\mathbf{h} = [\ 1.4 \quad 2.1\ ]$

$\boldsymbol{a} = [\ 2 \quad 3\ ]$

Scaled codewords added together plus noise.

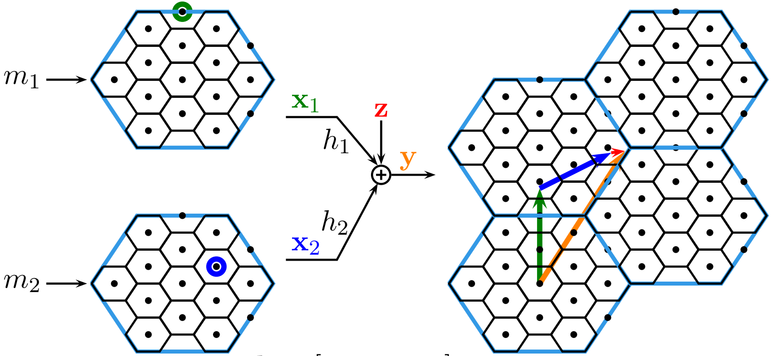


$$\mathbf{h} = [\ 1.4 \quad 2.1\ ]$$

$$\boldsymbol{a} = [\ 2 \quad 3\ ]$$

Compute-and-Forward: Illustration

Extra noise penalty for non-integer channel coefficients.



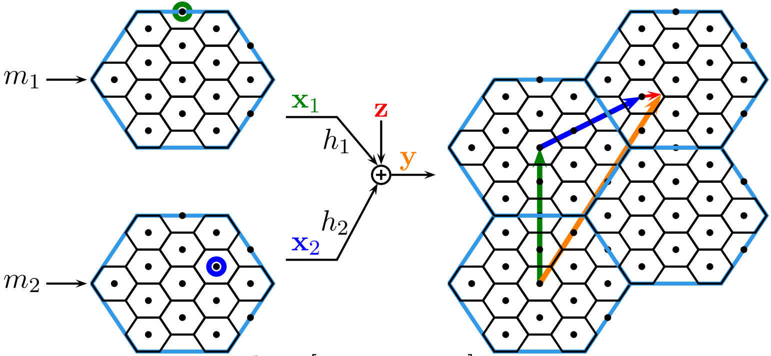
$$\mathbf{h} = [\ 1.4 \quad 2.1\ ]$$

$$\boldsymbol{a} = [\ 2 \quad 3\ ]$$

Effective noise: $1 + P\|\mathbf{h} - \mathbf{a}\|^2$

Compute-and-Forward: Illustration

Scale output by α to reduce non-integer noise penalty.



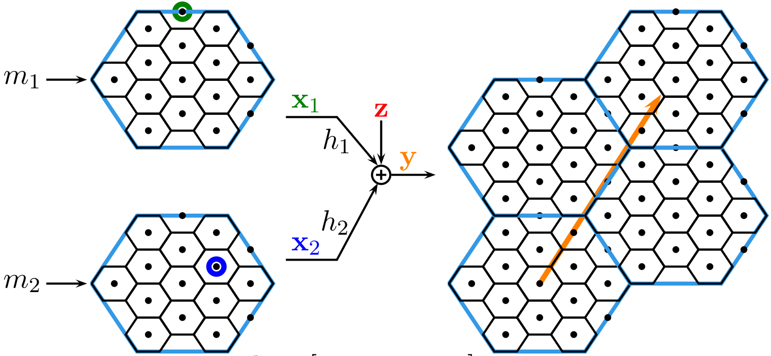
$$\alpha\mathbf{h} = [\ \alpha 1.4 \quad \alpha 2.1\ ]$$

$$\boldsymbol{a} = [\ 2 \quad 3\ ]$$

Effective noise: $\alpha^2 + P\|\alpha\mathbf{h} - \boldsymbol{a}\|^2$

Compute-and-Forward: Illustration

Scale output by α to reduce non-integer noise penalty.



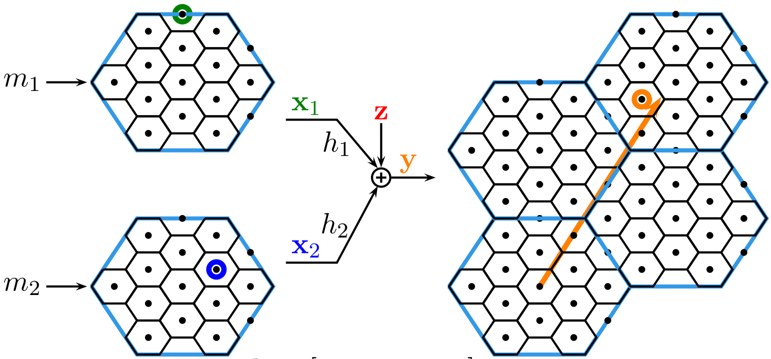
$$\alpha\mathbf{h} = [\ \alpha 1.4 \quad \alpha 2.1\ ]$$

$$\boldsymbol{a} = [\ 2 \quad 3\ ]$$

Effective noise: $\alpha^2 + P\|\alpha\mathbf{h} - \boldsymbol{a}\|^2$

Compute-and-Forward: Illustration

Decode to the closest lattice point.



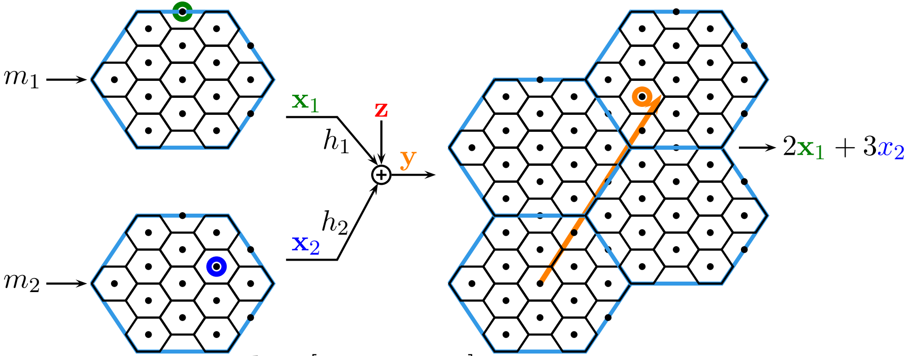
$$\alpha \mathbf{h} = [\ \alpha 1.4 \quad \alpha 2.1\ ]$$

$$\boldsymbol{a} = [\ 2 \quad 3\ ]$$

Effective noise: $\alpha^2 + P\|\alpha\mathbf{h} - \boldsymbol{a}\|^2$

Compute-and-Forward: Illustration

Recover integer linear combination of the codewords.



$$\alpha\mathbf{h} = [\ \alpha 1.4 \quad \alpha 2.1\ ]$$

$$\boldsymbol{a} = [\ 2 \qquad 3\ ]$$

Effective noise: $\alpha^2 + P\|\alpha\mathbf{h} - \boldsymbol{a}\|^2$

Compute-and-Forward: Achievable Rates

Theorem (Nazer-Gastpar '11)

A receiver can recover a linear combination with coefficient vector $\boldsymbol{a} \in \mathbb{Z}^K$ *over the channel vector* $\mathbf{h} \in \mathbb{R}^K$ *if* $R < R_{comp}(\mathbf{h}, \boldsymbol{a})$ *where*

$$R_{comp}(\mathbf{h}, \boldsymbol{a}) = \max_{\alpha \in \mathbb{R}} \frac{1}{2} \log^+ \left(\frac{P}{\alpha^2 + P \|\alpha \mathbf{h} - \boldsymbol{a}\|^2} \right).$$

Compute-and-Forward: Achievable Rates

Theorem (Nazer-Gastpar '11)

A receiver can recover a linear combination with coefficient vector $\boldsymbol{a} \in \mathbb{Z}^K$ *over the channel vector* $\mathbf{h} \in \mathbb{R}^K$ *if* $R < R_{comp}(\mathbf{h}, \boldsymbol{a})$ *where*

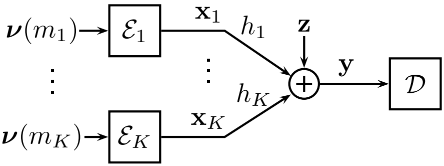
$$R_{comp}(\mathbf{h}, \boldsymbol{a}) = \frac{1}{2} \log^+ \left(\frac{P}{\boldsymbol{a}^\mathsf{T} \left(P^{-1} \mathbf{I} + \ \mathbf{h}\mathbf{h}^\mathsf{T} \right)^{-1} \boldsymbol{a}} \right).$$

Compute-and-Forward: Achievable Rates

Theorem (Nazer-Gastpar '11)

A receiver can recover a linear combination with coefficient vector $\boldsymbol{a} \in \mathbb{Z}^K$ over the channel vector $\mathbf{h} \in \mathbb{R}^K$ if $R < R_{comp}(\mathbf{h}, \boldsymbol{a})$ where

$$R_{comp}(\mathbf{h}, \boldsymbol{a}) = \frac{1}{2} \log^+ \left(\frac{P}{\boldsymbol{a}^\mathsf{T} \left(P^{-1}\mathbf{I} + \mathbf{h}\mathbf{h}^\mathsf{T}\right)^{-1} \boldsymbol{a}} \right).$$

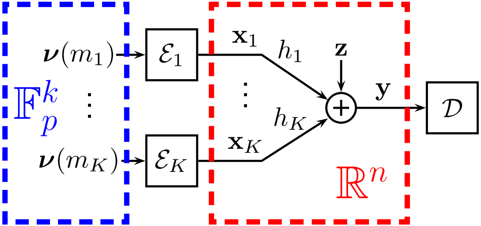


Compute-and-Forward: Achievable Rates

Theorem (Nazer-Gastpar '11)

A receiver can recover a linear combination with coefficient vector $\boldsymbol{a} \in \mathbb{Z}^K$ *over the channel vector* $\mathbf{h} \in \mathbb{R}^K$ *if* $R < R_{comp}(\mathbf{h}, \boldsymbol{a})$ *where*

$$R_{comp}(\mathbf{h}, \boldsymbol{a}) = \frac{1}{2} \log^+ \left(\frac{P}{\boldsymbol{a}^\mathsf{T} \left(P^{-1}\mathbf{I} + \mathbf{h}\mathbf{h}^\mathsf{T} \right)^{-1} \boldsymbol{a}} \right).$$

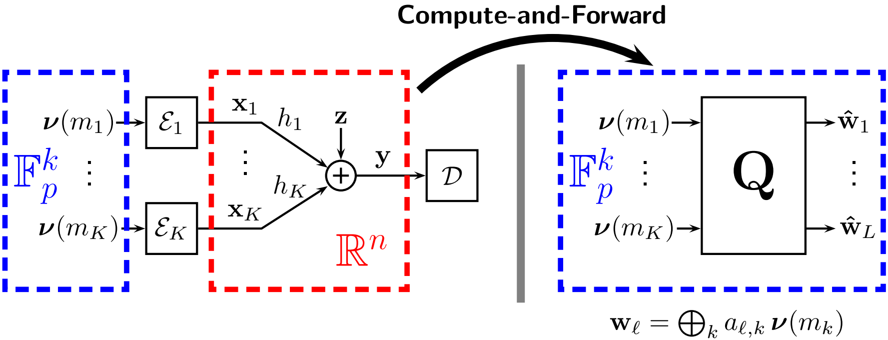


Compute-and-Forward: Achievable Rates

Theorem (Nazer-Gastpar '11)

A receiver can recover a linear combination with coefficient vector $\boldsymbol{a} \in \mathbb{Z}^K$ over the channel vector $\mathbf{h} \in \mathbb{R}^K$ if $R < R_{comp}(\mathbf{h}, \boldsymbol{a})$ where

$$R_{comp}(\mathbf{h}, \boldsymbol{a}) = \frac{1}{2} \log^+ \left(\frac{P}{\boldsymbol{a}^\mathsf{T} \left(P^{-1}\mathbf{I} + \ \mathbf{h}\mathbf{h}^\mathsf{T} \right)^{-1} \boldsymbol{a}} \right).$$



Theorem (Nazer-Gastpar '11)

A receiver can recover a linear combination with coefficient vector $\boldsymbol{a} \in \mathbb{Z}^K$ *over the channel vector* $\mathbf{h} \in \mathbb{R}^K$ *if* $R < R_{comp}(\mathbf{h}, \boldsymbol{a})$ *where*

$$R_{comp}(\mathbf{h}, \boldsymbol{a}) = \frac{1}{2} \log^+ \left(\frac{P}{\boldsymbol{a}^\mathsf{T} \left(P^{-1}\mathbf{I} + \mathbf{h}\mathbf{h}^\mathsf{T} \right)^{-1} \boldsymbol{a}} \right).$$

Special Cases:

- Perfect Match: $R_{\text{comp}}(\boldsymbol{a}, \boldsymbol{a}) = \frac{1}{2} \log^+ \left(\frac{1}{\|\boldsymbol{a}\|^2} + P \right)$

Compute-and-Forward: Achievable Rates

Theorem (Nazer-Gastpar '11)

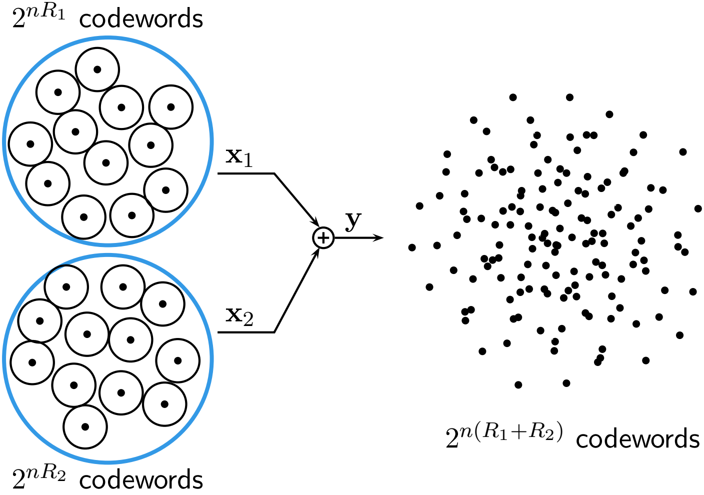
A receiver can recover a linear combination with coefficient vector $\boldsymbol{a} \in \mathbb{Z}^K$ over the channel vector $\mathbf{h} \in \mathbb{R}^K$ if $R < R_{comp}(\mathbf{h}, \boldsymbol{a})$ where

$$R_{comp}(\mathbf{h}, \boldsymbol{a}) = \frac{1}{2} \log^+ \left(\frac{P}{\boldsymbol{a}^\mathsf{T} \big(P^{-1}\mathbf{I} + \mathbf{h}\mathbf{h}^\mathsf{T}\big)^{-1} \boldsymbol{a}} \right).$$

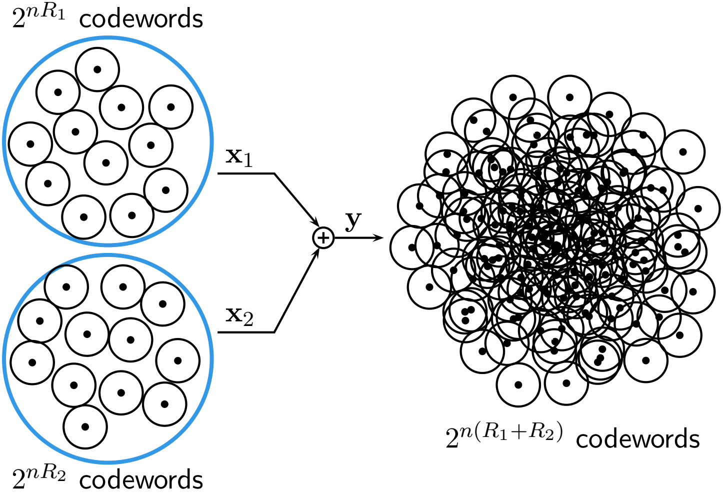
Special Cases:

- Perfect Match: $R_{\text{comp}}(\boldsymbol{a}, \boldsymbol{a}) = \frac{1}{2} \log^+ \left(\frac{1}{\|\boldsymbol{a}\|^2} + P \right)$
- Decode the k^{th} Message:
$$R_{\text{comp}}\Big(\mathbf{h}, [\underbrace{0 \ \cdots \ 0}_{k-1 \text{ zeros}} \ 1 \ 0 \ \cdots \ 0]^\mathsf{T}\Big) = \frac{1}{2} \log \left(1 + \frac{h_k^2 P}{1 + P \sum_{\ell \neq k} h_\ell^2} \right)$$

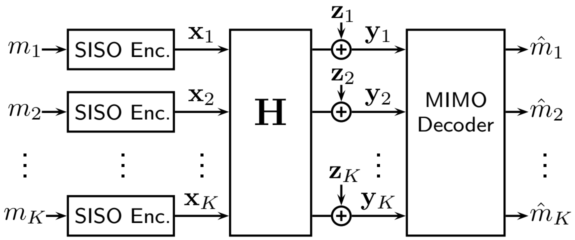
Random i.i.d. codes are not good for computation



Random i.i.d. codes are not good for computation



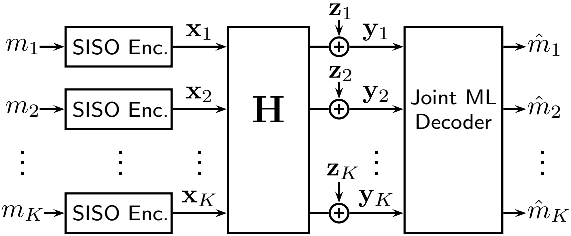
Application: MIMO Uplink Channel



Usual Assumptions:

- Each antenna carries an independent data stream $\mathbf{x}_\ell \in \mathbb{C}^n$ of rate R (e.g., V-BLAST setting, cellular uplink). $\mathbf{X} = [\mathbf{x}_1 \ \cdots \ \mathbf{x}_K]^\mathsf{T}$.
- Usual power constraint: $\|\mathbf{x}_\ell\|^2 \leq nP$.
- Channel model: $\mathbf{Y} = \mathbf{H}\mathbf{X} + \mathbf{Z}$
- $\mathbf{Z}$ is elementwise i.i.d. $\mathcal{CN}(0,1)$.
- CSIR: Only the receiver knows channel realization $\mathbf{H} \in \mathbb{C}^{K \times K}$.

MIMO Uplink Channel: Joint ML Decoding

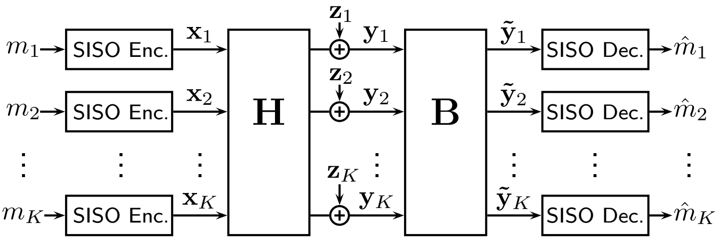


Joint Maximum Likelihood Decoding:

$$R_{\text{joint}}(\mathbf{H}) = \min_{\mathcal{S} \subseteq \{1,\ldots,K\}} \frac{1}{|\mathcal{S}|} \log\det\left(\mathbf{I} + P\ \mathbf{H}_{\mathcal{S}}\mathbf{H}_{\mathcal{S}}^*\right)$$

- Corresponds to the (symmetric) outage capacity.
- Naive implementation has prohibitively high complexity.
- Of course, there are many clever ways to reduce the complexity!

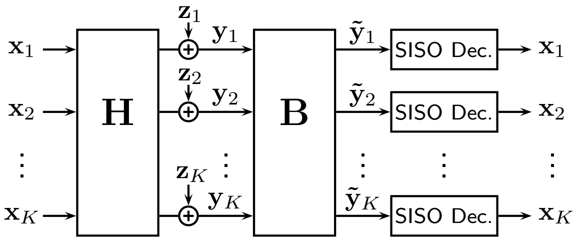
MIMO Uplink Channel: Zero-Forcing and Linear MMSE



Zero-Forcing and Linear MMSE Receivers:

- Project the received signal, $\tilde{\mathbf{Y}} = \mathbf{B}\mathbf{Y}$ to eliminate interference between data streams.
- After projection, single-user decoders attempt to recover the individual data streams.
- Optimal $\mathbf{B}$ is the MMSE projection.

MIMO Uplink Channel: Zero-Forcing and Linear MMSE



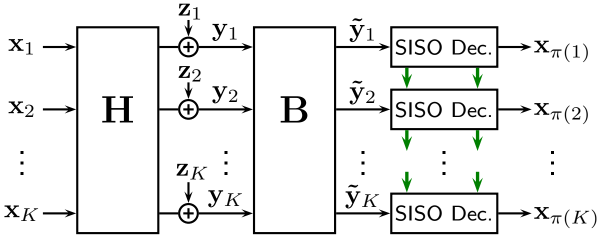
Zero-Forcing and Linear MMSE Receivers:

- The k^{th} SISO decoder tries to recover $\mathbf{x}_k$ from $\mathbf{b}_k^{\mathsf{T}}\mathbf{Y}$:

$$\mathsf{SINR}_{\mathsf{LMMSE},k}(\mathbf{H}) = \max_{\mathbf{b}_k} \frac{P\ \|\mathbf{b}_k^{\mathsf{T}}\mathbf{h}_k\|^2}{1 + P\ \sum_{\ell \neq k} \|\mathbf{b}_k^{\mathsf{T}}\mathbf{h}_\ell\|^2}$$

- Rate per user:

$$R_{\mathsf{LMMSE}}(\mathbf{H}) = \min_{k=1,\ldots,K} \log\left(1 + \mathsf{SINR}_{\mathsf{LMMSE},k}(\mathbf{H})\right)$$



Successive Interference Cancellation Receivers:

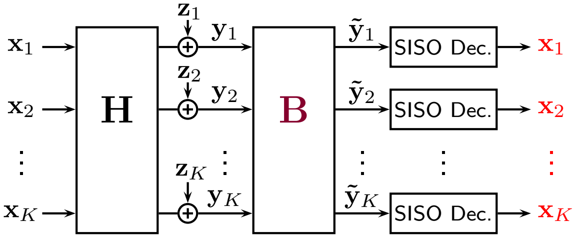
- Decode in order π. Cancel $\mathbf{x}_{\pi(1)}, \ldots, \mathbf{x}_{\pi(k-1)}$ from $\tilde{\mathbf{y}}_k$:

$$\mathsf{SINR}_{\mathsf{SIC},\pi(m)}(\mathbf{H}) = \max_{\mathbf{b}_m} \frac{P \, \|\mathbf{b}_k^{\mathsf{T}} \mathbf{h}_{\pi(k)}\|^2}{1 + \mathsf{SNR} \, \sum_{\ell=k+1}^{K} \|\mathbf{b}_k^{\mathsf{T}} \mathbf{h}_{\pi(\ell)}\|^2}$$

- Rate per user:

$$R_{\mathsf{V\text{-}BLAST\ II}}(\mathbf{H}) = \max_{\pi} \min_{k=1,\ldots,K} \log\left(1 + \mathsf{SINR}_{\mathsf{SIC},\pi(k)}(\mathbf{H})\right)$$

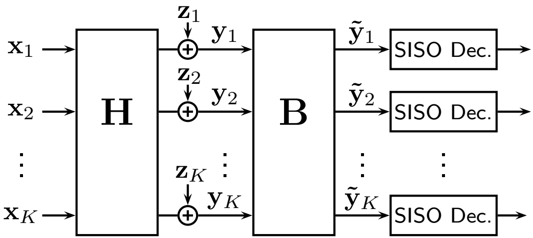
MIMO Uplink Channel: Integer-Forcing



What if we could decode something else?

- Zero-Forcing / LMMSE: First, eliminate interference.
  Then, decode individual data streams.

MIMO Uplink Channel: Integer-Forcing

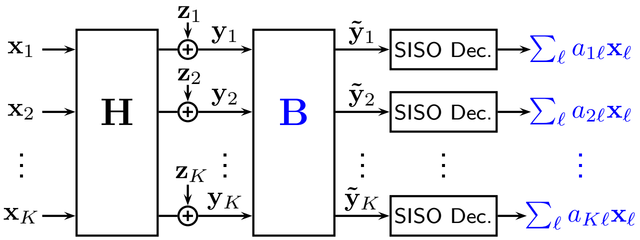


What if we could decode something else?

- Zero-Forcing / LMMSE: First, eliminate interference.
  Then, decode individual data streams.

First, decode

MIMO Uplink Channel: Integer-Forcing

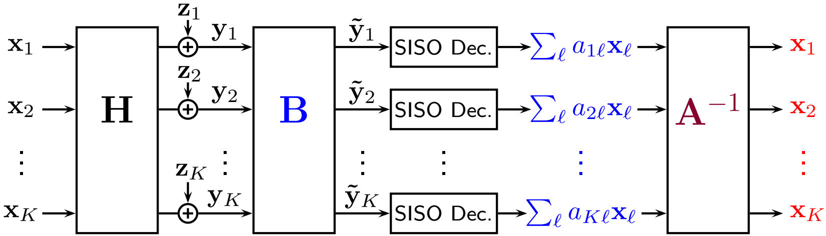


What if we could decode something else?

- Zero-Forcing / LMMSE: First, eliminate interference.
  Then, decode individual data streams.

- Integer-Forcing: First, decode integer-linear combinations.

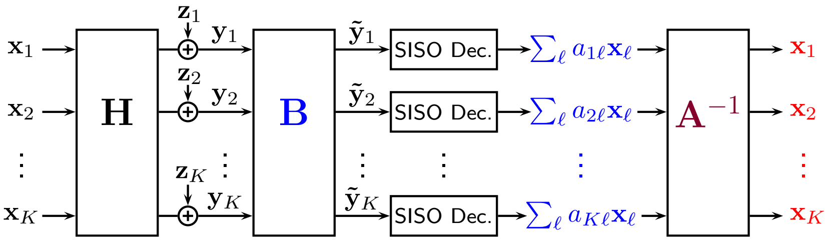
MIMO Uplink Channel: Integer-Forcing



What if we could decode something else?

- Zero-Forcing / LMMSE: First, eliminate interference.
  Then, decode individual data streams.

- Integer-Forcing: First, decode integer-linear combinations.
  Then, eliminate interference.



What if we could decode something else?

- Zero-Forcing / LMMSE: First, eliminate interference.
  Then, decode individual data streams.

- Integer-Forcing: First, decode integer-linear combinations.
  Then, eliminate interference.

- If the integer matrix $\mathbf{A}$ is full rank, we can successfully recover the individual data streams.

Integer-Forcing Linear Receivers:

- The k^{th} effective channel after projection is

$$\mathbf{b}_k^{\mathsf{T}}\mathbf{Y} = \mathbf{b}_k^{\mathsf{T}}\mathbf{H}\mathbf{X} + \mathbf{b}_k^{\mathsf{T}}\mathbf{Z}$$

Integer-Forcing Linear Receivers:

- The k^{th} effective channel after projection is

$$\begin{aligned}\mathbf{b}_k^{\mathsf{T}}\mathbf{Y} &= \mathbf{b}_k^{\mathsf{T}}\mathbf{H}\mathbf{X} + \mathbf{b}_k^{\mathsf{T}}\mathbf{Z} \\ &= \mathbf{a}_k^{\mathsf{T}}\mathbf{X} + (\mathbf{b}_k^{\mathsf{T}}\mathbf{H} - \mathbf{a}_k^{\mathsf{T}})\mathbf{X} + \mathbf{b}_k^{\mathsf{T}}\mathbf{Z}\end{aligned}$$

Integer-Forcing Linear Receivers:

- The k^{th} effective channel after projection is

$$
\begin{aligned}
\mathbf{b}_k^{\mathsf{T}}\mathbf{Y} &= \mathbf{b}_k^{\mathsf{T}}\mathbf{H}\mathbf{X} + \mathbf{b}_k^{\mathsf{T}}\mathbf{Z} \\
&= \mathbf{a}_k^{\mathsf{T}}\mathbf{X} + (\mathbf{b}_k^{\mathsf{T}}\mathbf{H} - \mathbf{a}_k^{\mathsf{T}})\mathbf{X} + \mathbf{b}_k^{\mathsf{T}}\mathbf{Z} \\
&= \underbrace{\sum_{\ell=1}^{K} a_{k\ell}\mathbf{x}_\ell^{\mathsf{T}}}_{\text{Codeword}} + \underbrace{(\mathbf{b}_k^{\mathsf{T}}\mathbf{H} - \mathbf{a}_k^{\mathsf{T}})\mathbf{X} + \mathbf{b}_k^{\mathsf{T}}\mathbf{Z}}_{\text{Effective Noise}}
\end{aligned}
$$

Integer-Forcing Linear Receivers:

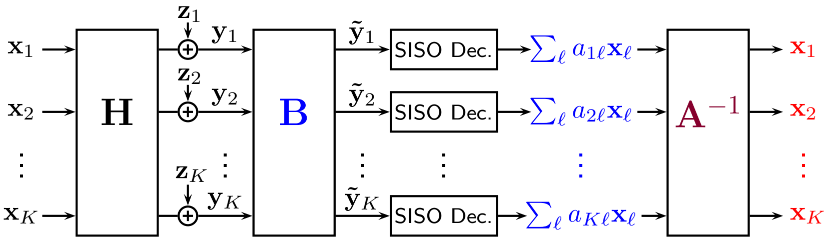
- The k^{th} effective channel after projection is

$$\begin{aligned} \mathbf{b}_k^{\mathsf{T}}\mathbf{Y} &= \mathbf{b}_k^{\mathsf{T}}\mathbf{H}\mathbf{X} + \mathbf{b}_k^{\mathsf{T}}\mathbf{Z} \\ &= \mathbf{a}_k^{\mathsf{T}}\mathbf{X} + (\mathbf{b}_k^{\mathsf{T}}\mathbf{H} - \mathbf{a}_k^{\mathsf{T}})\mathbf{X} + \mathbf{b}_k^{\mathsf{T}}\mathbf{Z} \\ &= \underbrace{\sum_{\ell=1}^{K} a_{k\ell}\mathbf{x}_\ell^{\mathsf{T}}}_{\text{Codeword}} + \underbrace{(\mathbf{b}_k^{\mathsf{T}}\mathbf{H} - \mathbf{a}_k^{\mathsf{T}})\mathbf{X} + \mathbf{b}_k^{\mathsf{T}}\mathbf{Z}}_{\text{Effective Noise}} \end{aligned}$$

- The $a_{k\ell} \in \mathbb{Z}[j]$ are Gaussian integers and the codebook should be closed under integer-linear combinations.

- We are free to choose any full-rank integer-valued matrix $\mathbf{A}$.

MIMO Uplink Channel: Integer-Forcing



Integer-Forcing Linear Receivers: (Zhan-Nazer-Erez-Gastpar '14)

- The k^{th} SISO decoder tries to recover $\sum_\ell a_{k\ell}\mathbf{x}_\ell$ from $\mathbf{b}_k^\mathsf{T}\mathbf{Y}$:

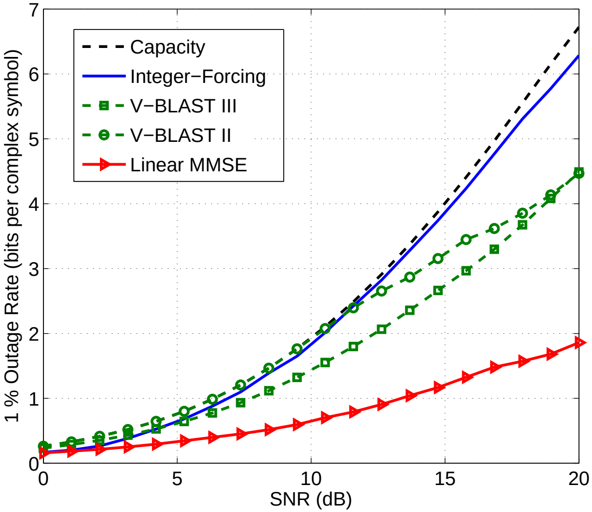
$$\mathsf{SINR}_{\mathsf{IF},k}(\mathbf{H},\mathbf{A}) = \max_{\mathbf{b}_k} \frac{P}{\|\mathbf{b}_k\|^2 + P\|\mathbf{b}_k^\mathsf{T}\mathbf{H} - \mathbf{a}_k^\mathsf{T}\|^2}$$

- Rate per user:

$$R_{\mathsf{IF}}(\mathbf{H}) = \max_{\mathbf{A}} \min_{k=1,\dots,K} \log^+\left(\mathsf{SINR}_{\mathsf{IF},k}(\mathbf{H},\mathbf{A})\right)$$

- Includes linear MMSE as a special case by setting $\mathbf{A} = \mathbf{I}$.

Comparison: Outage Rates



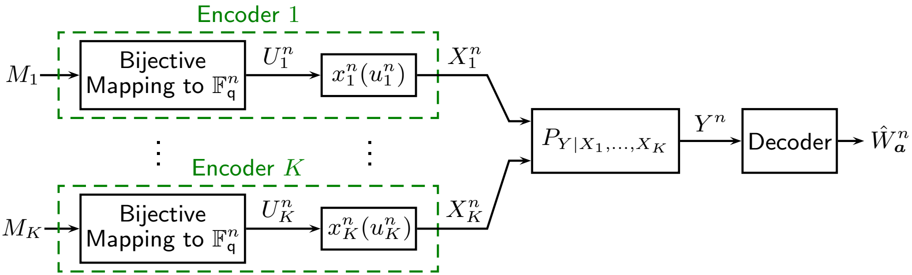
2 users, 2 receive antennas, Rayleigh fading, 1% outage.

Many Other Applications

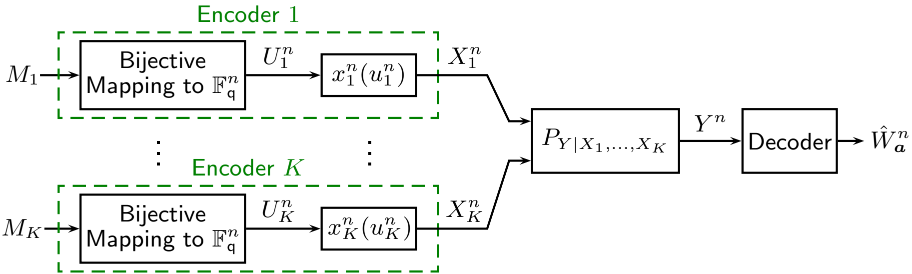
- Distributed Source Coding: **Körner-Marton '79**, **Krithivasan-Pradhan '09,'11**, **Wagner '11**, **Tse-Maddah-Ali '10**
- Relaying: **Wilson-Narayanan-Pfister-Sprintson '10**, **Nam-Chung-Lee '10, '11**, **Goseling-Gastpar-Weber '11**, **Song-Devroye '13**, **Nokleby-Aazhang '12**
- Cellular Networks: **Sanderovich-Peleg-Shamai '11**, **Nazer-Sanderovich-Gastpar-Shamai '09**, **Hong-Caire '13**
- Distributed Dirty-Paper Coding: **Philosof-Zamir '09**, **Philosof-Zamir-Erez-Khisti '11**, **Wang '12**
- Joint Source-Channel Coding: **Kochman-Zamir '09**, **Nazer-Gastpar '07, '08**, **Soundararajan-Vishwanath '12**
- Physical-Layer Secrecy: **He-Yener '11, '14**, **Kashyap-Shashank-Thangaraj '12**

- For the rest of the talk, I will discuss our recent efforts to bring these lattice coding ideas into the joint typicality framework.
- This is joint work with Sung Hoon Lim, Chen Feng, Adriano Pastore, and Michael Gastpar.
- See arXiv for our June 2016 pre-print.

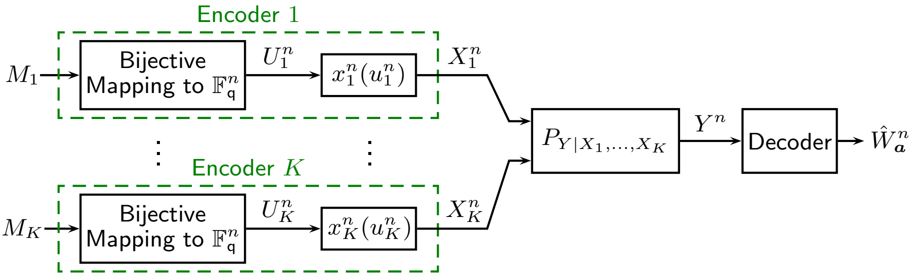
Compute-and-Forward: Beyond Gaussian Channels



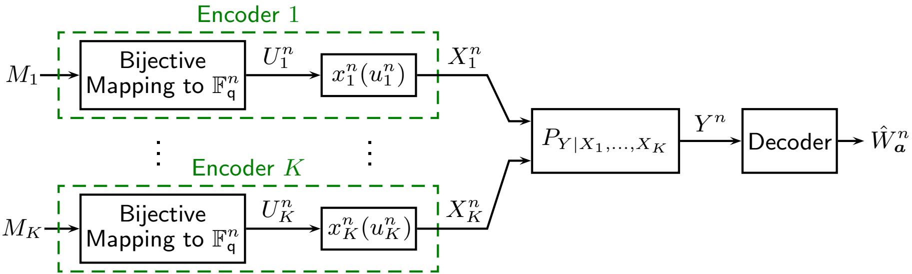
Compute-and-Forward: Beyond Gaussian Channels



- Messages: $m_k \in [2^{nR_k}] \triangleq \{0, \ldots, 2^{nR_k} - 1\}$, $k = 1, \ldots, K$.

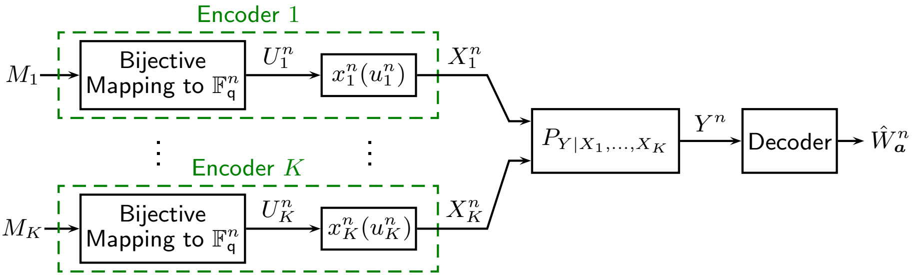


- Messages: $m_k \in [2^{nR_k}] \triangleq \{0, \ldots, 2^{nR_k} - 1\}$, $k = 1, \ldots, K$.
- Encoders: mappings $(u_k^n, x_k^n)(m_k) \in \mathbb{F}_q^n \times \mathcal{X}_k^n$, $k = 1, \ldots, K$ such that $u_k^n(m_k)$ is *bijective.*



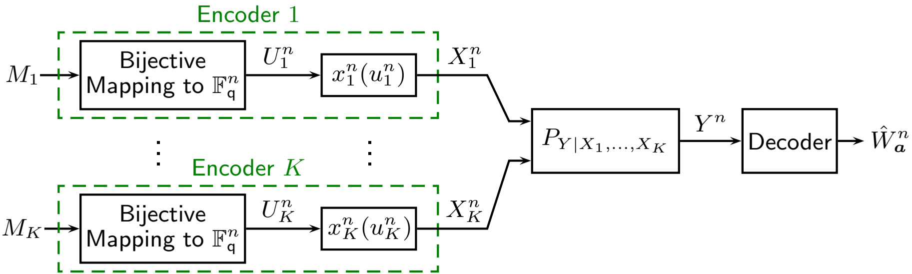
- Messages: $m_k \in [2^{nR_k}] \triangleq \{0, \ldots, 2^{nR_k} - 1\}$, $k = 1, \ldots, K$.
- Encoders: mappings $(u_k^n, x_k^n)(m_k) \in \mathbb{F}_\mathsf{q}^n \times \mathcal{X}_k^n$, $k = 1, \ldots, K$ such that $u_k^n(m_k)$ is *bijective.*
- Linear Combination: $w_{\boldsymbol{a}}^n \triangleq \bigoplus_k a_k u_k^n(m_k)$, $\boldsymbol{a} = [a_1 \ \cdots \ a_K] \in \mathbb{F}_\mathsf{q}^K$

Compute-and-Forward: Beyond Gaussian Channels



- Messages: $m_k \in [2^{nR_k}] \triangleq \{0, \ldots, 2^{nR_k} - 1\}$, $k = 1, \ldots, K$.
- Encoders: mappings $(u_k^n, x_k^n)(m_k) \in \mathbb{F}_q^n \times \mathcal{X}_k^n$, $k = 1, \ldots, K$ such that $u_k^n(m_k)$ is *bijective*.
- Linear Combination: $w_{\boldsymbol{a}}^n \triangleq \bigoplus_k a_k u_k^n(m_k)$, $\boldsymbol{a} = [a_1 \ \cdots \ a_K] \in \mathbb{F}_q^K$
- Decoder: assigns an estimate $\hat{w}_{\boldsymbol{a}}^n \in \mathbb{F}_q^n$ to each $y^n \in \mathcal{Y}^n$.

Compute-and-Forward: Beyond Gaussian Channels



- Messages: $m_k \in [2^{nR_k}] \triangleq \{0, \ldots, 2^{nR_k} - 1\}$, $k = 1, \ldots, K$.
- Encoders: mappings $(u_k^n, x_k^n)(m_k) \in \mathbb{F}_\mathsf{q}^n \times \mathcal{X}_k^n$, $k = 1, \ldots, K$ such that $u_k^n(m_k)$ is *bijective.*
- Linear Combination: $w_{\boldsymbol{a}}^n \triangleq \bigoplus_k a_k u_k^n(m_k)$, $\boldsymbol{a} = [a_1 \ \cdots \ a_K] \in \mathbb{F}_\mathsf{q}^K$
- Decoder: assigns an estimate $\hat{w}_{\boldsymbol{a}}^n \in \mathbb{F}_\mathsf{q}^n$ to each $y^n \in \mathcal{Y}^n$.
- Probability of Error: For uniformly distributed messages $M_1, \ldots, M_K$, want vanishing probability of error $\mathsf{P}\{\hat{W}_{\boldsymbol{a}}^n \neq W_{\boldsymbol{a}}^n\}$.

High-Level Intuition:

- Input Distribution: Want U_k^n to look typical with respect to pmf $p_{U_k}(u_k)$. There are $\approx 2^{nH(U_k)}$ typical sequences.

High-Level Intuition:

- Input Distribution: Want U_k^n to look typical with respect to pmf $p_{U_k}(u_k)$. There are $\approx 2^{nH(U_k)}$ typical sequences.

- True Codeword: Want $(W_{\boldsymbol{a}}^n, Y^n)$ to look jointly typical.

High-Level Intuition:

- Input Distribution: Want U_k^n to look typical with respect to pmf $p_{U_k}(u_k)$. There are $\approx 2^{nH(U_k)}$ typical sequences.

- True Codeword: Want $(W_{\boldsymbol{a}}^n, Y^n)$ to look jointly typical.

- Decoder searches for sequences $\tilde{w}_{\boldsymbol{a}}^n$ that are jointly typical with Y^n. There are $\approx 2^{nH(W_{\boldsymbol{a}}|Y)}$ possible sequences. If only one such sequence is jointly typical, declare it as the estimate $\hat{W}_{\boldsymbol{a}}^n$ of the linear combination $W_{\boldsymbol{a}}^n = a_1 U_1^n \oplus \cdots \oplus a_K U_K^n$.

High-Level Intuition:

- Input Distribution: Want U_k^n to look typical with respect to pmf $p_{U_k}(u_k)$. There are $\approx 2^{nH(U_k)}$ typical sequences.

- True Codeword: Want $(W_{\boldsymbol{a}}^n, Y^n)$ to look jointly typical.

- Decoder searches for sequences $\tilde{w}_{\boldsymbol{a}}^n$ that are jointly typical with Y^n. There are $\approx 2^{nH(W_{\boldsymbol{a}}|Y)}$ possible sequences. If only one such sequence is jointly typical, declare it as the estimate $\hat{W}_{\boldsymbol{a}}^n$ of the linear combination $W_{\boldsymbol{a}}^n = a_1 U_1^n \oplus \cdots \oplus a_K U_K^n$.

- We can show that, for this decoding strategy, we can achieve any rate tuple $(R_1, \ldots, R_K)$ satisfying

$$R_k < H(U_k) - H(W_{\boldsymbol{a}}|Y).$$

High-Level Intuition: (Low-Level Reality)

- Input Distribution: Want U_k^n to look typical with respect to pmf $p_{U_k}(u_k)$. There are $\approx 2^{nH(U_k)}$ typical sequences.

- True Codeword: Want $(W_{\boldsymbol{a}}^n, Y^n)$ to look jointly typical.

- Decoder searches for sequences $\tilde{w}_{\boldsymbol{a}}^n$ that are jointly typical with Y^n. There are $\approx 2^{nH(W_{\boldsymbol{a}}|Y)}$ possible sequences. If only one such sequence is jointly typical, declare it as the estimate $\hat{W}_{\boldsymbol{a}}^n$ of the linear combination $W_{\boldsymbol{a}}^n = a_1 U_1^n \oplus \cdots \oplus a_K U_K^n$.

- We can show that, for this decoding strategy, we can achieve any rate tuple $(R_1, \ldots, R_K)$ satisfying

$$R_k < H(U_k) - H(W_{\boldsymbol{a}}|Y).$$

High-Level Intuition: (Low-Level Reality)

- Input Distribution: Want U_k^n to look typical with respect to pmf $p_{U_k}(u_k)$. There are $\approx 2^{nH(U_k)}$ typical sequences.
  (Linear codewords look uniform.)
- True Codeword: Want $(W_{\boldsymbol{a}}^n, Y^n)$ to look jointly typical.
- Decoder searches for sequences $\tilde{w}_{\boldsymbol{a}}^n$ that are jointly typical with Y^n. There are $\approx 2^{nH(W_{\boldsymbol{a}}|Y)}$ possible sequences. If only one such sequence is jointly typical, declare it as the estimate $\hat{W}_{\boldsymbol{a}}^n$ of the linear combination $W_{\boldsymbol{a}}^n = a_1 U_1^n \oplus \cdots \oplus a_K U_K^n$.
- We can show that, for this decoding strategy, we can achieve any rate tuple $(R_1, \ldots, R_K)$ satisfying

$$R_k < H(U_k) - H(W_{\boldsymbol{a}}|Y).$$

High-Level Intuition: (Low-Level Reality)

- Input Distribution: Want U_k^n to look typical with respect to pmf $p_{U_k}(u_k)$. There are $\approx 2^{nH(U_k)}$ typical sequences.
(Linear codewords look uniform.)
- True Codeword: Want $(W_{\boldsymbol{a}}^n, Y^n)$ to look jointly typical.
(Proof is actually a bit involved.)
- Decoder searches for sequences $\tilde{w}_{\boldsymbol{a}}^n$ that are jointly typical with Y^n. There are $\approx 2^{nH(W_{\boldsymbol{a}}|Y)}$ possible sequences. If only one such sequence is jointly typical, declare it as the estimate $\hat{W}_{\boldsymbol{a}}^n$ of the linear combination $W_{\boldsymbol{a}}^n = a_1 U_1^n \oplus \cdots \oplus a_K U_K^n$.
- We can show that, for this decoding strategy, we can achieve any rate tuple $(R_1, \ldots, R_K)$ satisfying

$$R_k < H(U_k) - H(W_{\boldsymbol{a}}|Y).$$

High-Level Intuition: (Low-Level Reality)

- Input Distribution: Want U_k^n to look typical with respect to pmf $p_{U_k}(u_k)$. There are $\approx 2^{nH(U_k)}$ typical sequences.
(Linear codewords look uniform.)
- True Codeword: Want $(W_{\boldsymbol{a}}^n, Y^n)$ to look jointly typical.
(Proof is actually a bit involved.)
- Decoder searches for sequences $\tilde{w}_{\boldsymbol{a}}^n$ that are jointly typical with Y^n. There are $\approx 2^{nH(W_{\boldsymbol{a}}|Y)}$ possible sequences. If only one such sequence is jointly typical, declare it as the estimate $\hat{W}_{\boldsymbol{a}}^n$ of the linear combination $W_{\boldsymbol{a}}^n = a_1 U_1^n \oplus \cdots \oplus a_K U_K^n$.
(Suboptimal decoding rule)
- We can show that, for this decoding strategy, we can achieve any rate tuple $(R_1, \ldots, R_K)$ satisfying

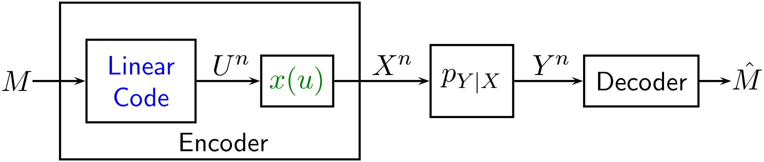
$$R_k < H(U_k) - H(W_{\boldsymbol{a}}|Y).$$

High-Level Intuition: (Low-Level Reality)

- Input Distribution: Want U_k^n to look typical with respect to pmf $p_{U_k}(u_k)$. There are $\approx 2^{nH(U_k)}$ typical sequences.
  (Linear codewords look uniform.)
- True Codeword: Want $(W_{\boldsymbol{a}}^n, Y^n)$ to look jointly typical.
  (Proof is actually a bit involved.)
- Decoder searches for sequences $\tilde{w}_{\boldsymbol{a}}^n$ that are jointly typical with Y^n. There are $\approx 2^{nH(W_{\boldsymbol{a}}|Y)}$ possible sequences. If only one such sequence is jointly typical, declare it as the estimate $\hat{W}_{\boldsymbol{a}}^n$ of the linear combination $W_{\boldsymbol{a}}^n = a_1 U_1^n \oplus \cdots \oplus a_K U_K^n$.
  (Suboptimal decoding rule)
- We can show that, for this decoding strategy, we can achieve any rate tuple $(R_1, \ldots, R_K)$ satisfying

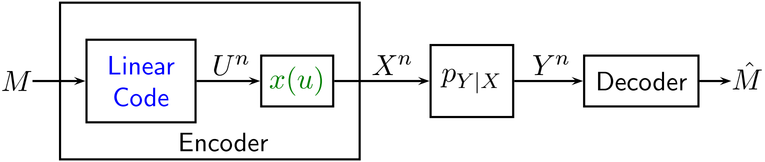
$$R_k < H(U_k) - H(W_{\boldsymbol{a}}|Y).$$

(Not a mutual information and can be negative.)



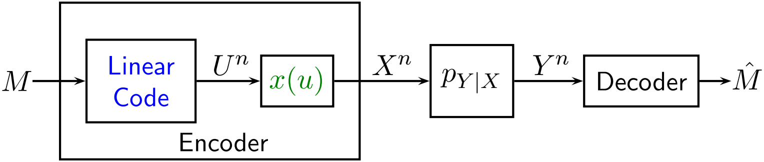
Code Construction:

Point-to-Point Channels: Linear Codes



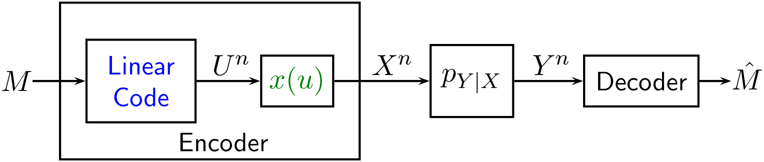
Code Construction:

- Pick a finite field $\mathbb{F}_{\mathsf{q}}$ and a symbol mapping $x : \mathbb{F}_{\mathsf{q}} \to \mathcal{X}$.



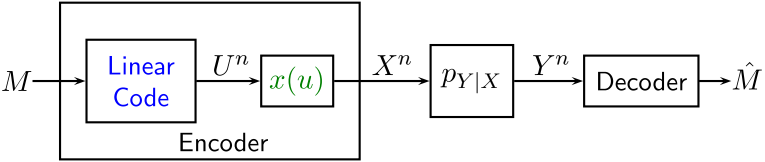
Code Construction:

- Pick a finite field $\mathbb{F}_{\mathsf{q}}$ and a symbol mapping $x : \mathbb{F}_{\mathsf{q}} \to \mathcal{X}$.
- Set $\kappa = nR/\log(\mathsf{q})$.



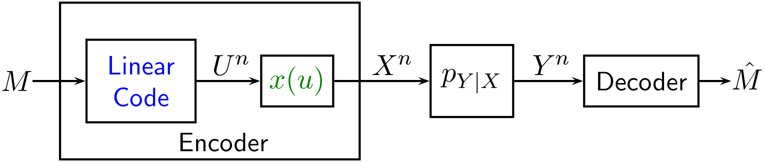
Code Construction:

- Pick a finite field $\mathbb{F}_\mathsf{q}$ and a symbol mapping $x : \mathbb{F}_\mathsf{q} \to \mathcal{X}$.
- Set $\kappa = nR/\log(\mathsf{q})$.
- Draw a random generator matrix $\mathbf{G} \in \mathbb{F}_\mathsf{q}^{\kappa \times n}$ elementwise i.i.d. $\mathrm{Unif}(\mathbb{F}_\mathsf{q})$. Let G be a realization.



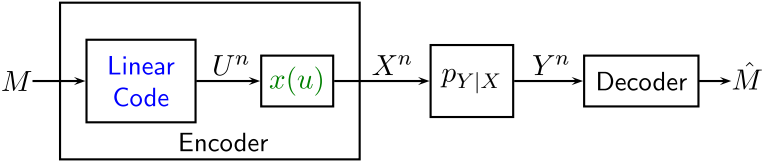
Code Construction:

- Pick a finite field $\mathbb{F}_\mathsf{q}$ and a symbol mapping $x : \mathbb{F}_\mathsf{q} \rightarrow \mathcal{X}$.
- Set $\kappa = nR / \log(\mathsf{q})$.
- Draw a random generator matrix $\mathbf{G} \in \mathbb{F}_\mathsf{q}^{\kappa \times n}$ elementwise i.i.d. $\mathrm{Unif}(\mathbb{F}_\mathsf{q})$. Let G be a realization.
- Draw a random shift (or "dither") D^n elementwise i.i.d. $\mathrm{Unif}(\mathbb{F}_\mathsf{q})$. Let d^n be a realization.



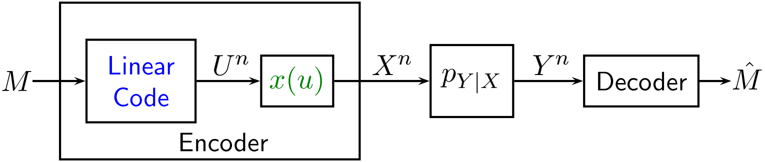
Code Construction:

- Pick a finite field $\mathbb{F}_\mathsf{q}$ and a symbol mapping $x : \mathbb{F}_\mathsf{q} \to \mathcal{X}$.
- Set $\kappa = nR/\log(\mathsf{q})$.
- Draw a random generator matrix $\mathbf{G} \in \mathbb{F}_\mathsf{q}^{\kappa \times n}$ elementwise i.i.d. $\mathrm{Unif}(\mathbb{F}_\mathsf{q})$. Let G be a realization.
- Draw a random shift (or "dither") D^n elementwise i.i.d. $\mathrm{Unif}(\mathbb{F}_\mathsf{q})$. Let d^n be a realization.
- Take q-ary expansion of message m into the vector $\boldsymbol{\nu}(m) \in \mathbb{F}_\mathsf{q}^\kappa$.



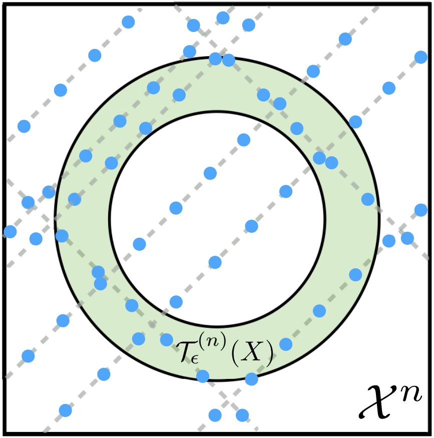
Code Construction:

- Pick a finite field $\mathbb{F}_{\mathsf{q}}$ and a symbol mapping $x : \mathbb{F}_{\mathsf{q}} \to \mathcal{X}$.
- Set $\kappa = nR/\log(\mathsf{q})$.
- Draw a random generator matrix $\mathbf{G} \in \mathbb{F}_{\mathsf{q}}^{\kappa \times n}$ elementwise i.i.d. $\mathrm{Unif}(\mathbb{F}_{\mathsf{q}})$. Let G be a realization.
- Draw a random shift (or "dither") D^n elementwise i.i.d. $\mathrm{Unif}(\mathbb{F}_{\mathsf{q}})$. Let d^n be a realization.
- Take q-ary expansion of message m into the vector $\boldsymbol{\nu}(m) \in \mathbb{F}_{\mathsf{q}}^{\kappa}$.
- Linear codeword for message m is $u^n(m) = \boldsymbol{\nu}(m)\mathsf{G} \oplus d^n$.



Code Construction:

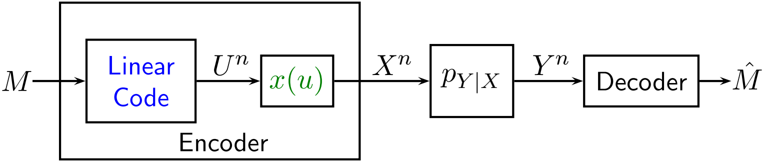
- Pick a finite field $\mathbb{F}_{\mathsf{q}}$ and a symbol mapping $x : \mathbb{F}_{\mathsf{q}} \to \mathcal{X}$.
- Set $\kappa = nR/\log(\mathsf{q})$.
- Draw a random generator matrix $\mathbf{G} \in \mathbb{F}_{\mathsf{q}}^{\kappa \times n}$ elementwise i.i.d. $\mathrm{Unif}(\mathbb{F}_{\mathsf{q}})$. Let G be a realization.
- Draw a random shift (or "dither") D^n elementwise i.i.d. $\mathrm{Unif}(\mathbb{F}_{\mathsf{q}})$. Let d^n be a realization.
- Take q-ary expansion of message m into the vector $\boldsymbol{\nu}(m) \in \mathbb{F}_{\mathsf{q}}^{\kappa}$.
- Linear codeword for message m is $u^n(m) = \boldsymbol{\nu}(m)\mathsf{G} \oplus d^n$.
- Channel input at time i is $x_i(m) = x(u_i(m))$.



Random Linear Codes

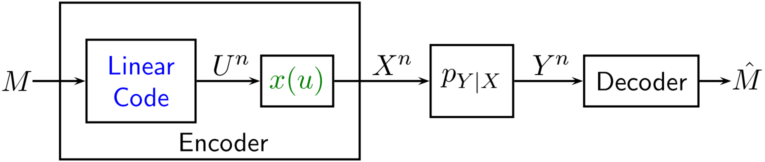
- Codewords are pairwise independent of one another.
- Codewords are uniformly distributed over $\mathbb{F}_q^n$.

Point-to-Point Channels: Linear Codes



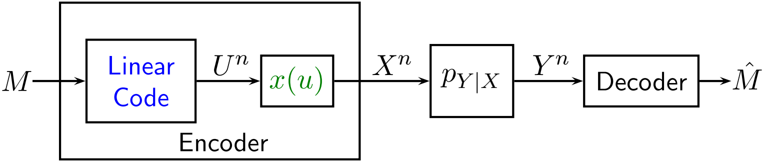
- Well known that a direct application of linear coding is not sufficient to reach the point-to-point capacity, **Ahlswede '71.**

Point-to-Point Channels: Linear Codes



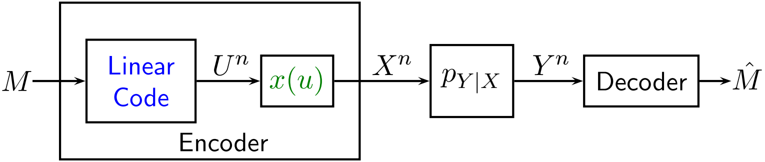
- Well known that a direct application of linear coding is not sufficient to reach the point-to-point capacity, **Ahlswede '71.**
- **Gallager '68:** Pick $\mathbb{F}_q$ with q $\gg \mathcal{X}$ and choose symbol mapping $x(u)$ to reach c.a.i.d. from $\mathrm{Unif}(\mathbb{F}_q)$. This can attain the capacity.

Point-to-Point Channels: Linear Codes

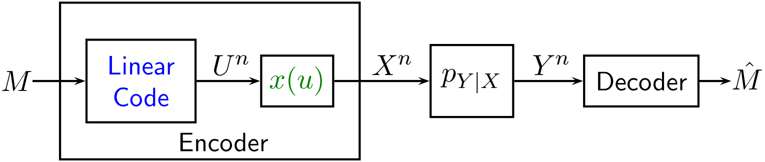


- Well known that a direct application of linear coding is not sufficient to reach the point-to-point capacity, **Ahlswede '71.**
- **Gallager '68:** Pick $\mathbb{F}_{\mathsf{q}}$ with q $\gg \mathcal{X}$ and choose symbol mapping $x(u)$ to reach c.a.i.d. from $\mathrm{Unif}(\mathbb{F}_{\mathsf{q}})$. This can attain the capacity.
- This will not work for us. Roughly speaking, if each encoder has a different input distribution, the symbol mappings may be quite different, which will disrupt the linear structure of the codebook.

Point-to-Point Channels: Linear Codes

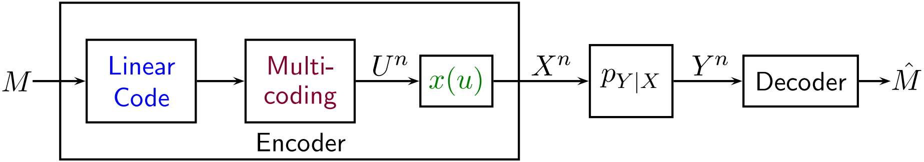


- Well known that a direct application of linear coding is not sufficient to reach the point-to-point capacity, **Ahlswede '71.**
- **Gallager '68:** Pick $\mathbb{F}_q$ with q $\gg \mathcal{X}$ and choose symbol mapping $x(u)$ to reach c.a.i.d. from $\mathrm{Unif}(\mathbb{F}_q)$. This can attain the capacity.
- This will not work for us. Roughly speaking, if each encoder has a different input distribution, the symbol mappings may be quite different, which will disrupt the linear structure of the codebook.
- **Padakandla-Pradhan '13:** It is possible to shape the input distribution using nested linear codes.

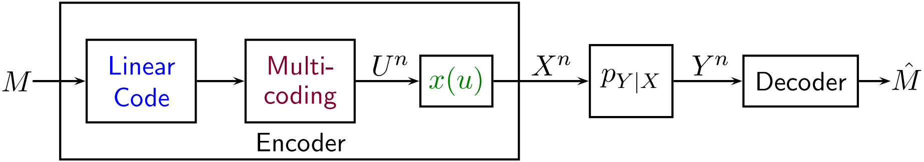


- Well known that a direct application of linear coding is not sufficient to reach the point-to-point capacity, **Ahlswede '71.**
- **Gallager '68:** Pick $\mathbb{F}_q$ with $q \gg \mathcal{X}$ and choose symbol mapping $x(u)$ to reach c.a.i.d. from $\mathrm{Unif}(\mathbb{F}_q)$. This can attain the capacity.
- This will not work for us. Roughly speaking, if each encoder has a different input distribution, the symbol mappings may be quite different, which will disrupt the linear structure of the codebook.
- **Padakandla-Pradhan '13:** It is possible to shape the input distribution using nested linear codes.
- Basic idea: Generate many codewords to represent one message. Search in this "bin" to find a codeword with the desired type, i.e., multicoding.

Point-to-Point Channels: Linear Codes + Multicoding

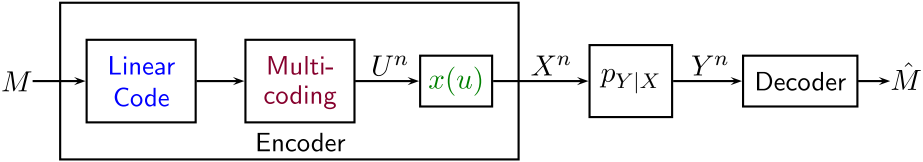


Code Construction:



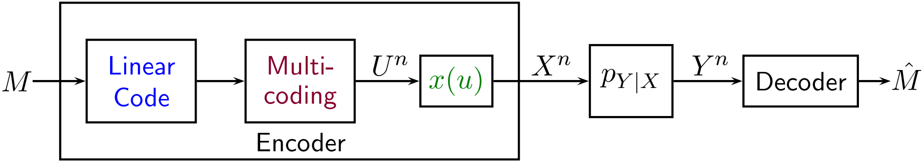
Code Construction:

- Messages $m \in [2^{nR}]$ and auxiliary indices $l \in [2^{n\hat{R}}]$.



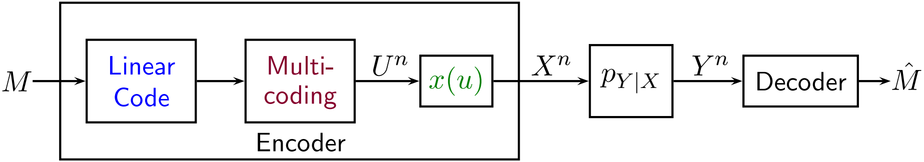
Code Construction:

- Messages $m \in [2^{nR}]$ and auxiliary indices $l \in [2^{n\hat{R}}]$.
- Set $\kappa = n(R + \hat{R})/\log(\mathsf{q})$.



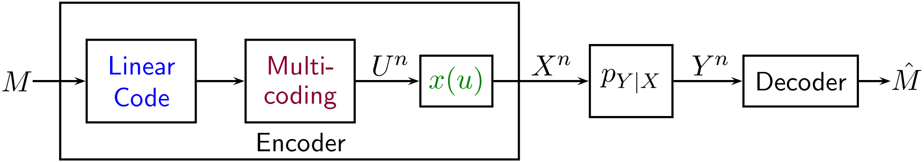
Code Construction:

- Messages $m \in [2^{nR}]$ and auxiliary indices $l \in [2^{n\hat{R}}]$.
- Set $\kappa = n(R + \hat{R})/\log(\mathsf{q})$.
- Pick generator matrix G and dither d^n as before.



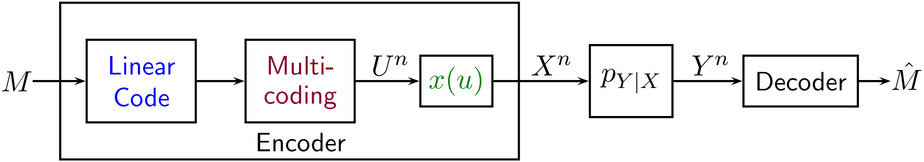
Code Construction:

- Messages $m \in [2^{nR}]$ and auxiliary indices $l \in [2^{n\hat{R}}]$.
- Set $\kappa = n(R + \hat{R}) / \log(\mathsf{q})$.
- Pick generator matrix G and dither d^n as before.
- Take q-ary expansions $\left[\boldsymbol{\nu}(m)\ \boldsymbol{\nu}(l)\right] \in \mathbb{F}_{\mathsf{q}}^{\kappa}$.



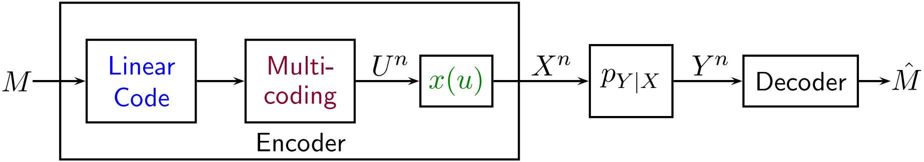
Code Construction:

- Messages $m \in [2^{nR}]$ and auxiliary indices $l \in [2^{n\hat{R}}]$.
- Set $\kappa = n(R + \hat{R}) / \log(\mathsf{q})$.
- Pick generator matrix G and dither d^n as before.
- Take q-ary expansions $\big[\boldsymbol{\nu}(m)\ \boldsymbol{\nu}(l)\big] \in \mathbb{F}_{\mathsf{q}}^{\kappa}$.
- Linear codewords: $u^n(m,l) = \big[\boldsymbol{\nu}(m)\ \boldsymbol{\nu}(l)\big]\,\mathsf{G} \oplus d^n$.


M
Linear Code
Multi-coding
U^n
$x(u)$
Encoder
X^n
$p_{Y|X}$
Y^n
Decoder
$\hat{M}$

Encoding:

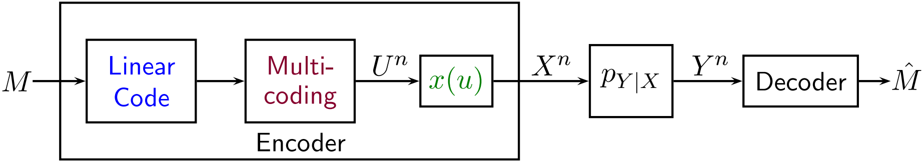
Point-to-Point Channels: Linear Codes + Multicoding



Encoding:

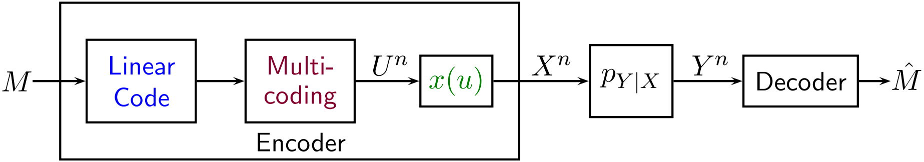
- Fix $p(u)$ and $x(u)$.

Point-to-Point Channels: Linear Codes + Multicoding



Encoding:

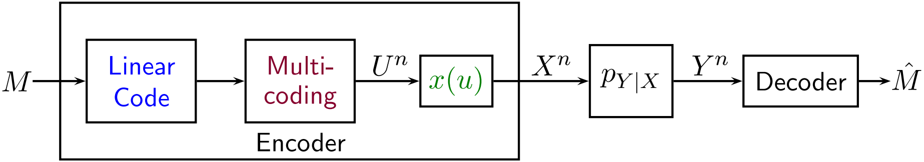
- Fix $p(u)$ and $x(u)$.
- Multicoding: For each m, find an index l such that $u^n(m,l) \in \mathcal{T}_{\epsilon'}^{(n)}(U)$



Encoding:

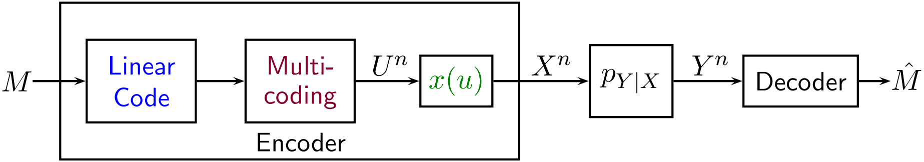
- Fix $p(u)$ and $x(u)$.
- Multicoding: For each m, find an index l such that $u^n(m,l) \in \mathcal{T}_{\epsilon'}^{(n)}(U)$
- Succeeds w.h.p. if $\hat{R} > D(p_U \| p_{\mathsf{q}})$ (where p_{q} is uniform over $\mathbb{F}_{\mathsf{q}}$).

Point-to-Point Channels: Linear Codes + Multicoding



Encoding:

- Fix $p(u)$ and $x(u)$.
- Multicoding: For each m, find an index l such that $u^n(m,l) \in \mathcal{T}_{\epsilon'}^{(n)}(U)$
- Succeeds w.h.p. if $\hat{R} > D(p_U \| p_{\mathsf{q}})$ (where p_{q} is uniform over $\mathbb{F}_{\mathsf{q}}$).
- Transmit $x_i = x\big(u_i(m,l)\big)$.

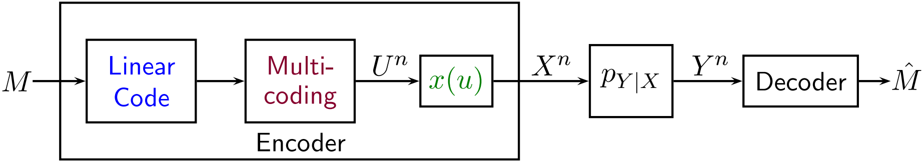


Encoding:

- Fix $p(u)$ and $x(u)$.
- Multicoding: For each m, find an index l such that $u^n(m,l) \in \mathcal{T}_{\epsilon'}^{(n)}(U)$
- Succeeds w.h.p. if $\hat{R} > D(p_U \| p_{\mathsf{q}})$ (where p_{q} is uniform over $\mathbb{F}_{\mathsf{q}}$).
- Transmit $x_i = x\big(u_i(m,l)\big)$.

Decoding:

Point-to-Point Channels: Linear Codes + Multicoding



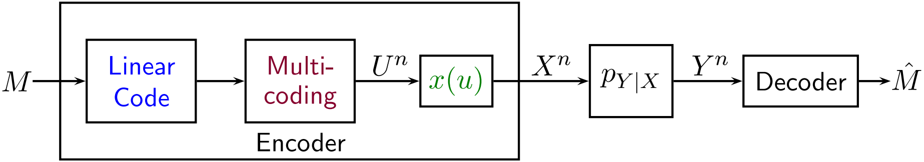
Encoding:

- Fix $p(u)$ and $x(u)$.
- Multicoding: For each m, find an index l such that $u^n(m,l) \in \mathcal{T}_{\epsilon'}^{(n)}(U)$
- Succeeds w.h.p. if $\hat{R} > D(p_U \| p_{\mathsf{q}})$ (where p_{q} is uniform over $\mathbb{F}_{\mathsf{q}}$).
- Transmit $x_i = x\big(u_i(m,l)\big)$.

Decoding:

- Joint Typicality Decoding: Find the unique index $\hat{m}$ such that $\big(u^n(\hat{m},\hat{l}),\, y^n\big) \in \mathcal{T}_{\epsilon}^{(n)}(U,Y)$ for some index $\hat{l}$.

Point-to-Point Channels: Linear Codes + Multicoding



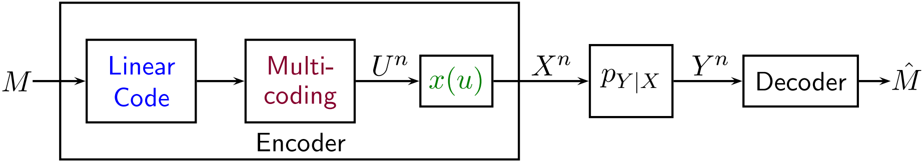
Encoding:

- Fix $p(u)$ and $x(u)$.
- Multicoding: For each m, find an index l such that $u^n(m,l) \in \mathcal{T}_{\epsilon'}^{(n)}(U)$
- Succeeds w.h.p. if $\hat{R} > D(p_U \| p_{\mathsf{q}})$ (where p_{q} is uniform over $\mathbb{F}_{\mathsf{q}}$).
- Transmit $x_i = x\big(u_i(m,l)\big)$.

Decoding:

- Joint Typicality Decoding: Find the unique index $\hat{m}$ such that $\big(u^n(\hat{m},\hat{l}),\, y^n\big) \in \mathcal{T}_{\epsilon}^{(n)}(U,Y)$ for some index $\hat{l}$.
- Succeeds w.h.p. if $R + \hat{R} < I(U;Y) + D(p_U \| p_{\mathsf{q}})$

Point-to-Point Channels: Linear Codes + Multicoding



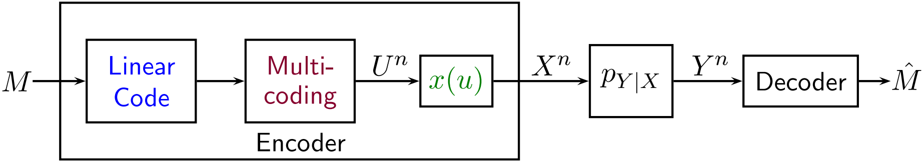
Theorem (Padakandla-Pradhan '13)

Any rate R satisfying

$$R < \max_{p(u),\, x(u)} I(U;Y)$$

is achievable. This is equal to the capacity if $\mathsf{q} \geq |\mathcal{X}|$.

Point-to-Point Channels: Linear Codes + Multicoding



Theorem (Padakandla-Pradhan '13)

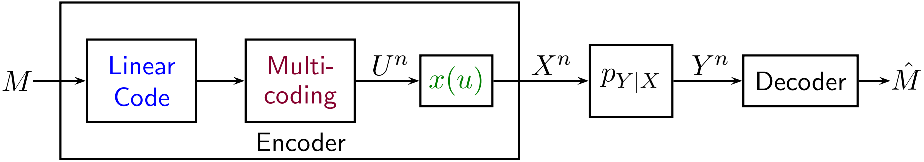
Any rate R satisfying

$$R < \max_{p(u),\, x(u)} I(U;Y)$$

is achievable. This is equal to the capacity if $\mathsf{q} \geq |\mathcal{X}|$.

- This is the basic coding framework that we will use for each transmitter.

Point-to-Point Channels: Linear Codes + Multicoding



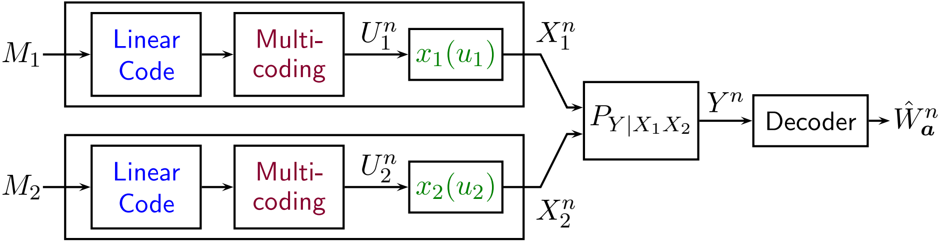
Theorem (Padakandla-Pradhan '13)

Any rate R satisfying

$$R < \max_{p(u),\, x(u)} I(U;Y)$$

is achievable. This is equal to the capacity if $\mathsf{q} \geq |\mathcal{X}|$.

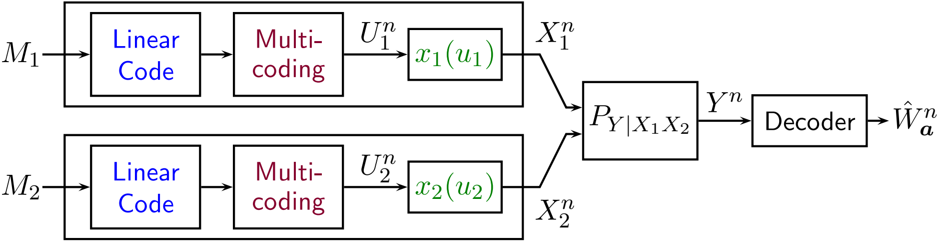
- This is the basic coding framework that we will use for each transmitter.
- Next, let's examine a two-transmitter, one-receiver "compute-and-forward" network.



Code Construction:

- Messages $m_k \in [2^{nR_k}]$ and auxiliary indices $l_k \in [2^{n\hat{R}_k}]$, $k = 1, 2$.

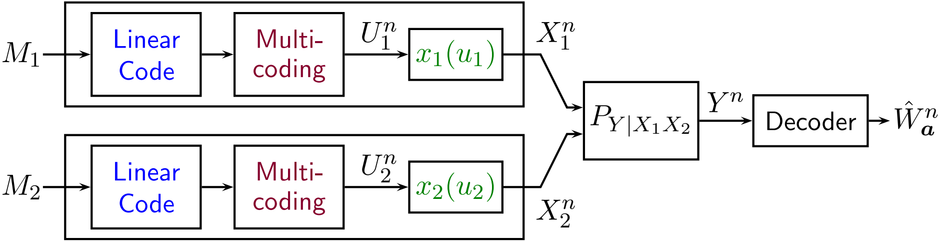
Nested Linear Coding Architecture



Code Construction:

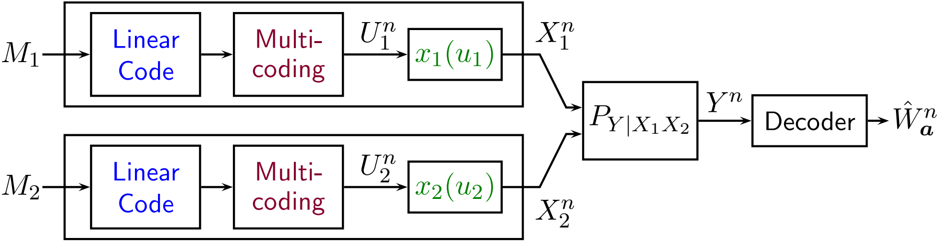
- Messages $m_k \in [2^{nR_k}]$ and auxiliary indices $l_k \in [2^{n\hat{R}_k}]$, $k = 1, 2$.
- Set $\kappa = n(\max\{R_1 + \hat{R}_1,\ R_2 + \hat{R}_2\})/\log(\mathsf{q})$.

Nested Linear Coding Architecture



Code Construction:

- Messages $m_k \in [2^{nR_k}]$ and auxiliary indices $l_k \in [2^{n\hat{R}_k}]$, $k = 1, 2$.
- Set $\kappa = n(\max\{R_1 + \hat{R}_1,\ R_2 + \hat{R}_2\}) / \log(\mathsf{q})$.
- Pick generator matrix G and dithers $d_1^n,\ d_2^n$ as before.

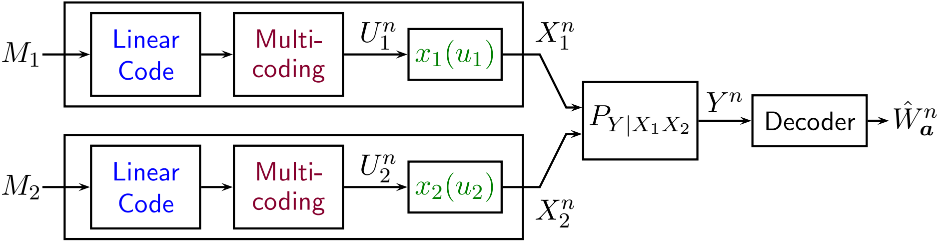


Code Construction:

- Messages $m_k \in [2^{nR_k}]$ and auxiliary indices $l_k \in [2^{n\hat{R}_k}]$, $k = 1, 2$.
- Set $\kappa = n(\max\{R_1 + \hat{R}_1,\ R_2 + \hat{R}_2\})/\log(\mathsf{q})$.
- Pick generator matrix G and dithers $d_1^n,\ d_2^n$ as before.
- Take q-ary expansions $\left[\boldsymbol{\nu}(m_1) \quad \boldsymbol{\nu}(l_1)\right] \in \mathbb{F}_{\mathsf{q}}^{\kappa}$

  $\left[\boldsymbol{\nu}(m_2)\ \boldsymbol{\nu}(l_2)\ \mathbf{0}\right] \in \mathbb{F}_{\mathsf{q}}^{\kappa}$ Zero-padding

Nested Linear Coding Architecture

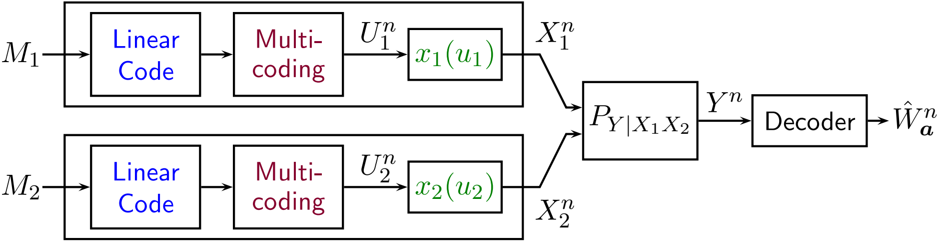


Code Construction:

- Messages $m_k \in [2^{nR_k}]$ and auxiliary indices $l_k \in [2^{n\hat{R}_k}]$, $k = 1, 2$.
- Set $\kappa = n(\max\{R_1 + \hat{R}_1,\ R_2 + \hat{R}_2\})/\log(\mathsf{q})$.
- Pick generator matrix G and dithers $d_1^n,\ d_2^n$ as before.
- Take q-ary expansions $\big[\boldsymbol{\eta}(m_1, l_1)\big] \in \mathbb{F}_{\mathsf{q}}^{\kappa}$

$$\big[\boldsymbol{\eta}(m_2, l_2)\big] \in \mathbb{F}_{\mathsf{q}}^{\kappa}$$

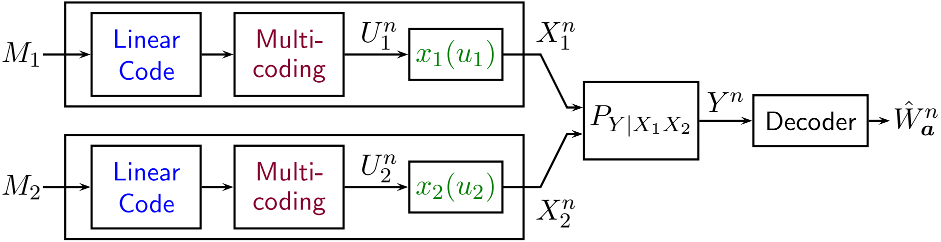
Nested Linear Coding Architecture



Code Construction:

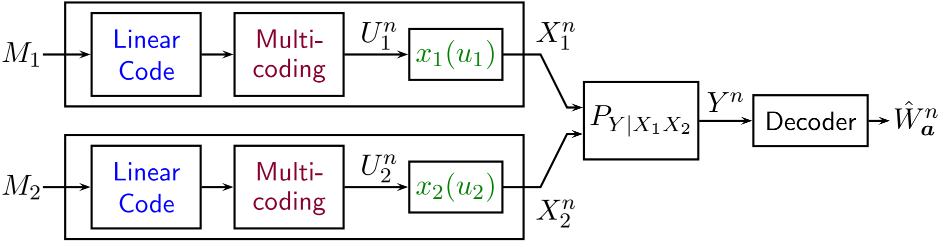
- Messages $m_k \in [2^{nR_k}]$ and auxiliary indices $l_k \in [2^{n\hat{R}_k}]$, $k = 1, 2$.
- Set $\kappa = n(\max\{R_1 + \hat{R}_1,\ R_2 + \hat{R}_2\})/\log(\mathsf{q})$.
- Pick generator matrix G and dithers $d_1^n,\ d_2^n$ as before.
- Take q-ary expansions $\big[\boldsymbol{\eta}(m_1, l_1)\big] \in \mathbb{F}_{\mathsf{q}}^{\kappa}$
  $\big[\boldsymbol{\eta}(m_2, l_2)\big] \in \mathbb{F}_{\mathsf{q}}^{\kappa}$
- Linear codewords: $u_1^n(m_1, l_1) = \boldsymbol{\eta}(m_1, l_1)\mathsf{G} \oplus d_1^n$
  $u_2^n(m_2, l_2) = \boldsymbol{\eta}(m_2, l_2)\mathsf{G} \oplus d_2^n$

Nested Linear Coding Architecture



Encoding:

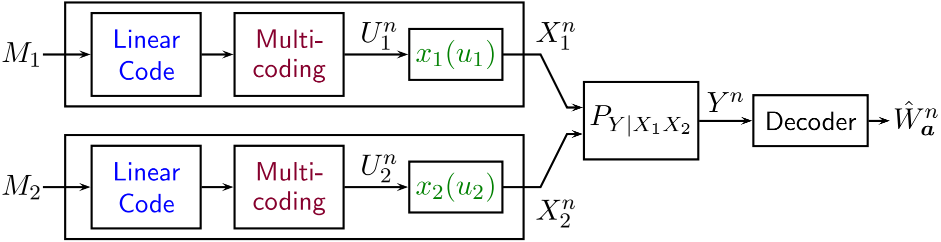
Nested Linear Coding Architecture



Encoding:

- Fix pmfs $p(u_1)$, $p(u_2)$ and mappings $x_1(u_1)$, and $x_2(u_2)$.

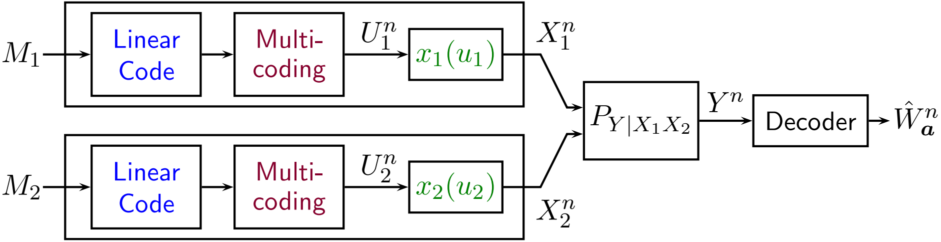
Nested Linear Coding Architecture



Encoding:

- Fix pmfs $p(u_1)$, $p(u_2)$ and mappings $x_1(u_1)$, and $x_2(u_2)$.
- Multicoding: For each m_k, find an index l_k such that $u_k^n(m_k, l_k) \in \mathcal{T}_{\epsilon'}^{(n)}(U_k)$. (If no such l_k, pick one randomly.)

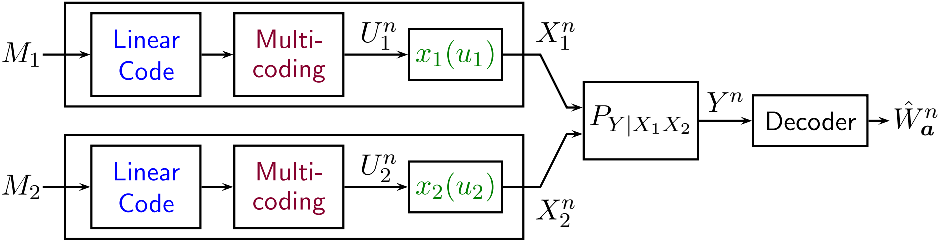
Nested Linear Coding Architecture



Encoding:

- Fix pmfs $p(u_1)$, $p(u_2)$ and mappings $x_1(u_1)$, and $x_2(u_2)$.
- Multicoding: For each m_k, find an index l_k such that $u_k^n(m_k, l_k) \in \mathcal{T}_{\epsilon'}^{(n)}(U_k)$. (If no such l_k, pick one randomly.)
- Transmit $x_{ki} = x_k\big(u_{ki}(m_k, l_k)\big), \;\; i = 1, \ldots, n.$

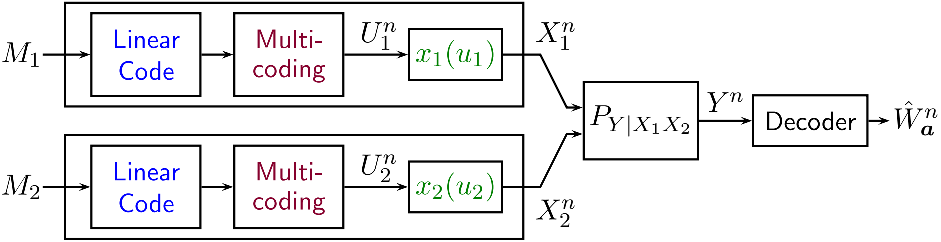
Nested Linear Coding Architecture



Encoding:

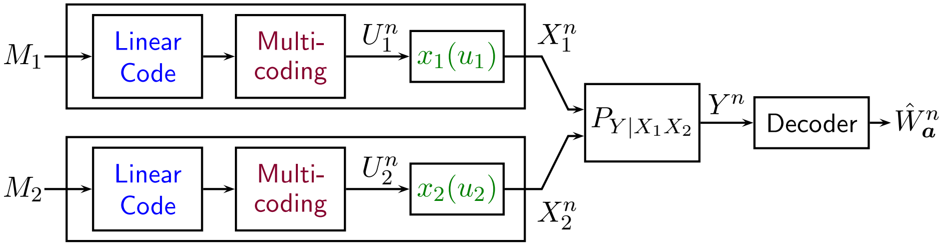
- Fix pmfs $p(u_1)$, $p(u_2)$ and mappings $x_1(u_1)$, and $x_2(u_2)$.
- Multicoding: For each m_k, find an index l_k such that $u_k^n(m_k, l_k) \in \mathcal{T}_{\epsilon'}^{(n)}(U_k)$. (If no such l_k, pick one randomly.)
- Transmit $x_{ki} = x_k\big(u_{ki}(m_k, l_k)\big), \;\; i = 1, \ldots, n.$

Nested Linear Coding Architecture



Computation Problem:

Nested Linear Coding Architecture



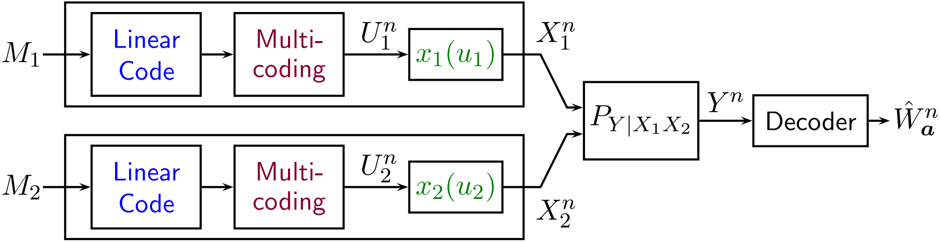
Computation Problem:

- For $m_k \in [2^{nR_k}]$, $l_k \in [2^{n\hat{R}_k}]$, we can express the linear combination of codewords as

$$\begin{aligned} w_{\boldsymbol{a}}^n &= a_1 u_1^n(m_1, l_1) \oplus a_2 u_2^n(m_2, l_2) \\ &= \big[a_1 \boldsymbol{\eta}(m_1, l_1) \oplus a_2 \boldsymbol{\eta}(m_2, l_2)\big] \mathsf{G} \oplus a_1 d_1^n \oplus a_2 d_2^n \\ &= \boldsymbol{\nu}(s_{\boldsymbol{a}}) \mathsf{G} \oplus a_1 d_1^n \oplus a_2 d_2^n \end{aligned}$$

where $s_{\boldsymbol{a}} \in [2^{n \max\{R_1+\hat{R}_1, R_2+\hat{R}_2\}}]$.

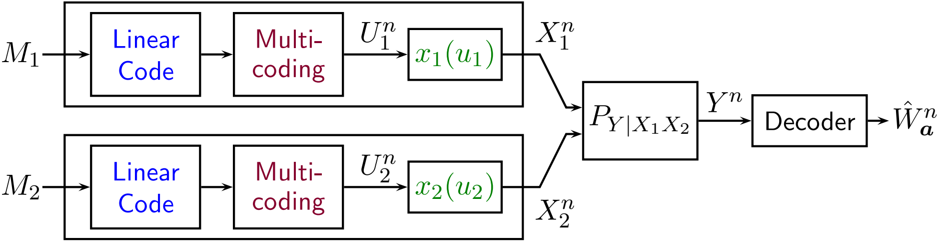
Nested Linear Coding Architecture



Decoding:

- Let $\epsilon' < \epsilon$.

Nested Linear Coding Architecture



Decoding:

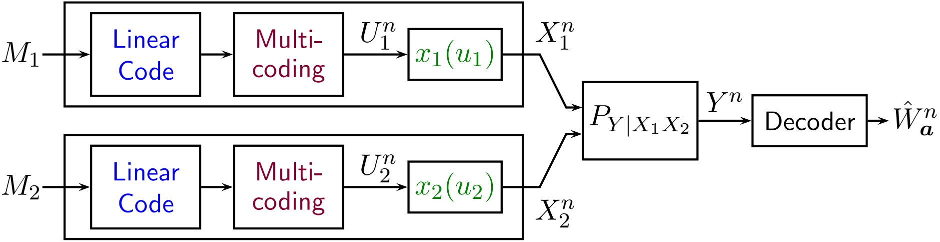
- Let $\epsilon' < \epsilon$.
- Search for a unique index $s_{\boldsymbol{a}} \in [2^{n\max\{R_1+\hat{R}_1, R_2+\hat{R}_2\}}]$ such that

$$(u_1^n(m_1, l_1), u_2^n(m_2, l_2), y^n) \in \mathcal{T}_\epsilon^{(n)}(U_1, U_2, Y),$$

for some $(m_1, l_1, m_2, l_2) \in [2^{nR_1}] \times [2^{n\hat{R}_1}] \times [2^{nR_2}] \times [2^{n\hat{R}_2}]$ such that

$$\boldsymbol{\nu}(s_{\boldsymbol{a}}) = a_1\boldsymbol{\eta}(m_1, l_1) \oplus a_2\boldsymbol{\eta}(m_2, l_2).$$

Nested Linear Coding Architecture



Decoding:

- Let $\epsilon' < \epsilon$.
- Search for a unique index $s_{\boldsymbol{a}} \in [2^{n\max\{R_1+\hat{R}_1, R_2+\hat{R}_2\}}]$ such that

$$(u_1^n(m_1,l_1), u_2^n(m_2,l_2), y^n) \in \mathcal{T}_\epsilon^{(n)}(U_1,U_2,Y),$$

  for some $(m_1,l_1,m_2,l_2) \in [2^{nR_1}] \times [2^{n\hat{R}_1}] \times [2^{nR_2}] \times [2^{n\hat{R}_2}]$ such that

$$\boldsymbol{\nu}(s_{\boldsymbol{a}}) = a_1\boldsymbol{\eta}(m_1,l_1) \oplus a_2\boldsymbol{\eta}(m_2,l_2).$$

- If there is no such index, or more than one, the decoder declares an error.

An error occurs only if one or more of the following events occur,

- For some message, we cannot find a typical linear codeword:

$$\mathcal{E}_1 = \{U_k^n(m_k, l_k) \notin \mathcal{T}_{\epsilon'}^{(n)} \text{ for all } l_k, \text{ for some } m_k, k = 1, 2\}.$$

Error Analysis

An error occurs only if one or more of the following events occur,

- For some message, we cannot find a typical linear codeword:

$$\mathcal{E}_1 = \{U_k^n(m_k, l_k) \notin \mathcal{T}_{\epsilon'}^{(n)} \text{ for all } l_k, \text{ for some } m_k, k = 1, 2\}.$$

- The channel inputs and output are not jointly typical:

$$\mathcal{E}_2 = \{(U_1^n(M_1, L_1), U_2^n(M_2, L_2), Y^n) \notin \mathcal{T}_{\epsilon}^{(n)}\}.$$

Error Analysis

An error occurs only if one or more of the following events occur,

- For some message, we cannot find a typical linear codeword:

$$\mathcal{E}_1 = \{U_k^n(m_k, l_k) \notin \mathcal{T}_{\epsilon'}^{(n)} \text{ for all } l_k, \text{ for some } m_k, k = 1, 2\}.$$

- The channel inputs and output are not jointly typical:

$$\mathcal{E}_2 = \{(U_1^n(M_1, L_1), U_2^n(M_2, L_2), Y^n) \notin \mathcal{T}_{\epsilon}^{(n)}\}.$$

- There are linear codewords that are jointly typical with the channel output and give the wrong linear combination:

$$\mathcal{E}_3 = \{(U_1^n(m_1, l_1), U_2^n(m_2, l_2), Y^n) \in \mathcal{T}_{\epsilon}^{(n)} \text{ for some } (m_1, l_1, m_2, l_2) \\ \text{such that } \boldsymbol{\nu}(S_{\boldsymbol{a}}) \neq a_1\boldsymbol{\eta}(m_1, l_1) \oplus a_2\boldsymbol{\eta}(m_2, l_2)\}.$$

An error occurs only if one or more of the following events occur,

- For some message, we cannot find a typical linear codeword:

$$\mathcal{E}_1 = \{U_k^n(m_k, l_k) \notin \mathcal{T}_{\epsilon'}^{(n)} \text{ for all } l_k, \text{ for some } m_k, k = 1, 2\}.$$

- The channel inputs and output are not jointly typical:

$$\mathcal{E}_2 = \{(U_1^n(M_1, L_1), U_2^n(M_2, L_2), Y^n) \notin \mathcal{T}_{\epsilon}^{(n)}\}.$$

- There are linear codewords that are jointly typical with the channel output and give the wrong linear combination:

$$\mathcal{E}_3 = \{(U_1^n(m_1, l_1), U_2^n(m_2, l_2), Y^n) \in \mathcal{T}_{\epsilon}^{(n)} \text{ for some } (m_1, l_1, m_2, l_2) \text{ such that } \boldsymbol{\nu}(S_{\boldsymbol{a}}) \neq a_1 \boldsymbol{\eta}(m_1, l_1) \oplus a_2 \boldsymbol{\eta}(m_2, l_2)\}.$$

Then, by the union of events bound,

$$\mathsf{P}\{\hat{W}_{\boldsymbol{a}}^n \neq W_{\boldsymbol{a}}^n\} \leq \mathsf{P}\{\mathcal{E}_1\} + \mathsf{P}\{\mathcal{E}_2 \cap \mathcal{E}_1^c\} + \mathsf{P}\{\mathcal{E}_3 \cap \mathcal{E}_1^c\}.$$

Error Analysis

- For some message, we cannot find a typical linear codeword:

$$\mathcal{E}_1 = \{U_k^n(m_k, l_k) \notin \mathcal{T}_{\epsilon'}^{(n)} \text{ for all } l_k, \text{ for some } m_k, k = 1, 2\}.$$

Error Analysis

- For some message, we cannot find a typical linear codeword:

$$\mathcal{E}_1 = \{U_k^n(m_k, l_k) \notin \mathcal{T}_{\epsilon'}^{(n)} \text{ for all } l_k, \text{ for some } m_k, k = 1, 2\}.$$

- If $\hat{R}_k > D(p_{U_k} \| p_{\mathsf{q}}) + \delta(\epsilon)$, then $\lim_{n\to\infty} \mathsf{P}\{\mathcal{E}_1\} = 0$ where $\delta(\epsilon) \to 0$ as $\epsilon \to 0$.

Error Analysis

- For some message, we cannot find a typical linear codeword:

$$\mathcal{E}_1 = \{U_k^n(m_k, l_k) \notin \mathcal{T}_{\epsilon'}^{(n)} \text{ for all } l_k, \text{ for some } m_k, k = 1, 2\}.$$

- If $\hat{R}_k > D(p_{U_k} \| p_{\mathsf{q}}) + \delta(\epsilon)$, then $\lim_{n\to\infty} \mathsf{P}\{\mathcal{E}_1\} = 0$ where $\delta(\epsilon) \to 0$ as $\epsilon \to 0$.
- $D(p_{U_k} \| p_{\mathsf{q}}) = \log \mathsf{q} - H(U_k)$.

Error Analysis

- For some message, we cannot find a typical linear codeword:

$$\mathcal{E}_1 = \{U_k^n(m_k, l_k) \notin \mathcal{T}_{\epsilon'}^{(n)} \text{ for all } l_k, \text{ for some } m_k, k = 1, 2\}.$$

- If $\hat{R}_k > D(p_{U_k} \| p_{\mathsf{q}}) + \delta(\epsilon)$, then $\lim_{n\to\infty} \mathsf{P}\{\mathcal{E}_1\} = 0$ where $\delta(\epsilon) \to 0$ as $\epsilon \to 0$.
- $D(p_{U_k} \| p_{\mathsf{q}}) = \log \mathsf{q} - H(U_k)$.
- Intuition: Searching for one of $\approx 2^{nH(U_k)}$ typical sequences out of $2^{n \log q}$ total sequences. Will succeed w.h.p. if $2^{n\hat{R}_k} > 2^{n(\log q - H(U_k))}$.

Error Analysis

- For some message, we cannot find a typical linear codeword:

$$\mathcal{E}_1 = \{U_k^n(m_k, l_k) \notin \mathcal{T}_{\epsilon'}^{(n)} \text{ for all } l_k, \text{ for some } m_k, k = 1, 2\}.$$

- If $\hat{R}_k > D(p_{U_k} \| p_{\mathsf{q}}) + \delta(\epsilon)$, then $\lim_{n\to\infty} \mathsf{P}\{\mathcal{E}_1\} = 0$ where $\delta(\epsilon) \to 0$ as $\epsilon \to 0$.
- $D(p_{U_k} \| p_{\mathsf{q}}) = \log \mathsf{q} - H(U_k)$.
- Intuition: Searching for one of $\approx 2^{nH(U_k)}$ typical sequences out of $2^{n \log q}$ total sequences. Will succeed w.h.p. if $2^{n\hat{R}_k} > 2^{n(\log q - H(U_k))}$.
- Proof just requires second moment method.

Error Analysis

- The channel inputs and output are not jointly typical:

$$\mathcal{E}_2 = \{(U_1^n(M_1, L_1), U_2^n(M_2, L_2), Y^n) \notin \mathcal{T}_\epsilon^{(n)}\}.$$

Error Analysis

- The channel inputs and output are not jointly typical:

$$\mathcal{E}_2 = \{(U_1^n(M_1, L_1), U_2^n(M_2, L_2), Y^n) \notin \mathcal{T}_\epsilon^{(n)}\}.$$

- If $\hat{R}_k > D(p_{U_k} \| p_{\mathsf{q}}) + \delta(\epsilon)$, then $\lim_{n\to\infty} \mathsf{P}\{\mathcal{E}_2 \cap \mathcal{E}_1^c\} = 0$ where $\delta(\epsilon) \to 0$ as $\epsilon \to 0$.

- The channel inputs and output are not jointly typical:

$$\mathcal{E}_2 = \{(U_1^n(M_1, L_1), U_2^n(M_2, L_2), Y^n) \notin \mathcal{T}_\epsilon^{(n)}\}.$$

- If $\hat{R}_k > D(p_{U_k} \| p_{\mathsf{q}}) + \delta(\epsilon)$, then $\lim_{n\to\infty} \mathsf{P}\{\mathcal{E}_2 \cap \mathcal{E}_1^c\} = 0$ where $\delta(\epsilon) \to 0$ as $\epsilon \to 0$.
- In a random i.i.d. coding proof, we would just use the fact that the codewords are independent and that the channel is memoryless.

Error Analysis

- The channel inputs and output are not jointly typical:

$$\mathcal{E}_2 = \{(U_1^n(M_1, L_1), U_2^n(M_2, L_2), Y^n) \notin \mathcal{T}_\epsilon^{(n)}\}.$$

- If $\hat{R}_k > D(p_{U_k} \| p_{\mathsf{q}}) + \delta(\epsilon)$, then $\lim_{n\to\infty} \mathsf{P}\{\mathcal{E}_2 \cap \mathcal{E}_1^c\} = 0$ where $\delta(\epsilon) \to 0$ as $\epsilon \to 0$.
- In a random i.i.d. coding proof, we would just use the fact that the codewords are independent and that the channel is memoryless.
- Here, the linear codewords can be statistically dependent, since the choices of the auxiliary indices L_k is coupled due to the shared nested linear codebook.

Error Analysis

- The channel inputs and output are not jointly typical:

$$\mathcal{E}_2 = \{(U_1^n(M_1, L_1), U_2^n(M_2, L_2), Y^n) \notin \mathcal{T}_\epsilon^{(n)}\}.$$

- If $\hat{R}_k > D(p_{U_k} \| p_{\mathsf{q}}) + \delta(\epsilon)$, then $\lim_{n\to\infty} \mathsf{P}\{\mathcal{E}_2 \cap \mathcal{E}_1^c\} = 0$ where $\delta(\epsilon) \to 0$ as $\epsilon \to 0$.
- In a random i.i.d. coding proof, we would just use the fact that the codewords are independent and that the channel is memoryless.
- Here, the linear codewords can be statistically dependent, since the choices of the auxiliary indices L_k is coupled due to the shared nested linear codebook.
- Our proof handles these statistical dependencies by breaking up the possible error events according to the underlying rank of the selected linear codewords. (Markov Lemma for Nested Linear Codes.)

Error Analysis

- The channel inputs and output are not jointly typical:

$$\mathcal{E}_2 = \{(U_1^n(M_1, L_1), U_2^n(M_2, L_2), Y^n) \notin \mathcal{T}_\epsilon^{(n)}\}.$$

- If $\hat{R}_k > D(p_{U_k} \| p_{\mathsf{q}}) + \delta(\epsilon)$, then $\lim_{n\to\infty} \mathsf{P}\{\mathcal{E}_2 \cap \mathcal{E}_1^c\} = 0$ where $\delta(\epsilon) \to 0$ as $\epsilon \to 0$.
- In a random i.i.d. coding proof, we would just use the fact that the codewords are independent and that the channel is memoryless.
- Here, the linear codewords can be statistically dependent, since the choices of the auxiliary indices L_k is coupled due to the shared nested linear codebook.
- Our proof handles these statistical dependencies by breaking up the possible error events according to the underlying rank of the selected linear codewords. (Markov Lemma for Nested Linear Codes.)
- Prior work by **Padakandla-Pradhan '13** developed a bound that also requires $\hat{R}_k < D(p_{U_k} \| p_{\mathsf{q}}) + 3\delta(\epsilon)$.

- There are linear codewords that are jointly typical with the channel output and give the wrong linear combination:

$$\mathcal{E}_3 = \{(U_1^n(m_1,l_1), U_2^n(m_2,l_2), Y^n) \in \mathcal{T}_\epsilon^{(n)} \text{ for some } (m_1,l_1,m_2,l_2) \\ \text{such that } \boldsymbol{\nu}(S_{\boldsymbol{a}}) \neq a_1\boldsymbol{\eta}(m_1,l_1) \oplus a_2\boldsymbol{\eta}(m_2,l_2)\}.$$

- There are linear codewords that are jointly typical with the channel output and give the wrong linear combination:

$$\mathcal{E}_3 = \{(U_1^n(m_1, l_1), U_2^n(m_2, l_2), Y^n) \in \mathcal{T}_\epsilon^{(n)} \text{ for some } (m_1, l_1, m_2, l_2) \\ \text{such that } \boldsymbol{\nu}(S_{\boldsymbol{a}}) \neq a_1 \boldsymbol{\eta}(m_1, l_1) \oplus a_2 \boldsymbol{\eta}(m_2, l_2)\}.$$

- We upper bound this event in two ways.

- There are linear codewords that are jointly typical with the channel output and give the wrong linear combination:

$$\mathcal{E}_3 = \{(U_1^n(m_1, l_1), U_2^n(m_2, l_2), Y^n) \in \mathcal{T}_\epsilon^{(n)} \text{ for some } (m_1, l_1, m_2, l_2) \text{ such that } \boldsymbol{\nu}(S_{\boldsymbol{a}}) \neq a_1 \boldsymbol{\eta}(m_1, l_1) \oplus a_2 \boldsymbol{\eta}(m_2, l_2)\}.$$

- We upper bound this event in two ways.
  1. "Direct Decoding" Bound

$$\mathsf{P}\{\mathcal{E}_3 \cap \mathcal{E}_1^c\} \leq \mathsf{P}\Big\{(W_{\boldsymbol{a}}^n(s_{\boldsymbol{a}}), Y^n) \in \mathcal{T}_\epsilon^{(n)},\ \mathcal{E}_1^c,\ s_{\boldsymbol{a}} \neq S_{\boldsymbol{a}}\Big\}$$

Error Analysis

- There are linear codewords that are jointly typical with the channel output and give the wrong linear combination:

$$\mathcal{E}_3 = \{(U_1^n(m_1,l_1), U_2^n(m_2,l_2), Y^n) \in \mathcal{T}_\epsilon^{(n)} \text{ for some } (m_1,l_1,m_2,l_2) \\ \text{such that } \boldsymbol{\nu}(S_{\boldsymbol{a}}) \neq a_1\boldsymbol{\eta}(m_1,l_1) \oplus a_2\boldsymbol{\eta}(m_2,l_2)\}.$$

- We upper bound this event in two ways.
  1. "Direct Decoding" Bound

$$\mathsf{P}\{\mathcal{E}_3 \cap \mathcal{E}_1^c\} \leq \mathsf{P}\Big\{(W_{\boldsymbol{a}}^n(s_{\boldsymbol{a}}), Y^n) \in \mathcal{T}_\epsilon^{(n)},\ \mathcal{E}_1^c,\ s_{\boldsymbol{a}} \neq S_{\boldsymbol{a}}\Big\}$$

  2. "Multiple-Access Decoding" Bound

$$\mathsf{P}\{\mathcal{E}_3 \cap \mathcal{E}_1^c\} \leq \mathsf{P}\Big\{(U_1^n(m_1,l_1), U_2^n(m_2,l_2), Y^n) \in \mathcal{T}_\epsilon^{(n)},\ \mathcal{E}_1^c \\ \text{for some } (m_1,l_1,m_2,l_2) \neq (M_1,L_1,M_2,L_2)\Big\}$$

Error Analysis: "Direct Decoding" Bound

$$\mathsf{P}\{\mathcal{E}_3 \cap \mathcal{E}_1^c\} \leq \mathsf{P}\Big\{(W_{\boldsymbol{a}}^n(s_{\boldsymbol{a}}), Y^n) \in \mathcal{T}_\epsilon^{(n)},\ \mathcal{E}_1^c,\ s_{\boldsymbol{a}} \neq S_{\boldsymbol{a}}\Big\}$$

Error Analysis: "Direct Decoding" Bound

$$\mathsf{P}\{\mathcal{E}_3 \cap \mathcal{E}_1^c\} \leq \mathsf{P}\Big\{(W_{\boldsymbol{a}}^n(s_{\boldsymbol{a}}), Y^n) \in \mathcal{T}_\epsilon^{(n)},\ \mathcal{E}_1^c,\ s_{\boldsymbol{a}} \neq S_{\boldsymbol{a}}\Big\}$$

- Can show that $\lim_{n\to\infty} \mathsf{P}\{\mathcal{E}_3 \cap \mathcal{E}_1^c\} = 0$ if

$$R_1 < I_{\mathsf{CF},1}(\boldsymbol{a}) \triangleq H(U_1) - H(W_{\boldsymbol{a}}|Y)$$
$$R_2 < I_{\mathsf{CF},2}(\boldsymbol{a}) \triangleq H(U_2) - H(W_{\boldsymbol{a}}|Y),$$

which matches our intuition from earlier.

Error Analysis: "Multiple-Access Decoding" Bound

$$\mathsf{P}\{\mathcal{E}_3 \cap \mathcal{E}_1^c\} \le \mathsf{P}\Big\{(U_1^n(m_1,l_1), U_2^n(m_2,l_2), Y^n) \in \mathcal{T}_\epsilon^{(n)},\ \mathcal{E}_1^c \text{ for some } (m_1,l_1,m_2,l_2) \ne (M_1,L_1,M_2,L_2)\Big\}$$

Error Analysis: "Multiple-Access Decoding" Bound

$$\mathsf{P}\{\mathcal{E}_3 \cap \mathcal{E}_1^c\} \le \mathsf{P}\Big\{(U_1^n(m_1,l_1), U_2^n(m_2,l_2), Y^n) \in \mathcal{T}_\epsilon^{(n)},\ \mathcal{E}_1^c \text{ for some } (m_1,l_1,m_2,l_2) \ne (M_1,L_1,M_2,L_2)\Big\}$$

- Can show that $\lim_{n\to\infty} \mathsf{P}\{\mathcal{E}_3 \cap \mathcal{E}_1^c\} = 0$ if

$$\begin{aligned} R_1 &< \max_{\boldsymbol{b}\in\mathbb{A}^2\setminus\{\boldsymbol{0}\}} \min\{I_{\mathsf{CF},1}(\boldsymbol{b}), I(X_1,X_2;Y) - I_{\mathsf{CF},2}(\boldsymbol{b})\}, \\ R_2 &< I(X_2;Y|X_1), \\ R_1 + R_2 &< I(X_1,X_2;Y) \end{aligned}$$

Error Analysis: "Multiple-Access Decoding" Bound

$$\mathsf{P}\{\mathcal{E}_3 \cap \mathcal{E}_1^c\} \le \mathsf{P}\Big\{(U_1^n(m_1,l_1), U_2^n(m_2,l_2), Y^n) \in \mathcal{T}_\epsilon^{(n)},\ \mathcal{E}_1^c \text{ for some } (m_1,l_1,m_2,l_2) \ne (M_1,L_1,M_2,L_2)\Big\}$$

- Can show that $\lim_{n\to\infty} \mathsf{P}\{\mathcal{E}_3 \cap \mathcal{E}_1^c\} = 0$ if

$$\begin{aligned} R_1 &< \max_{\boldsymbol{b} \in \mathbb{A}^2 \setminus \{\mathbf{0}\}} \min\{I_{\mathsf{CF},1}(\boldsymbol{b}), I(X_1,X_2;Y) - I_{\mathsf{CF},2}(\boldsymbol{b})\}, \\ R_2 &< I(X_2;Y|X_1), \\ R_1 + R_2 &< I(X_1,X_2;Y) \end{aligned}$$

OR

$$\begin{aligned} R_1 &< I(X_1;Y|X_2), \\ R_2 &< \max_{\boldsymbol{b} \in \mathbb{A}^2 \setminus \{\mathbf{0}\}} \min\{I_{\mathsf{CF},2}(\boldsymbol{b}), I(X_1,X_2;Y) - I_{\mathsf{CF},1}(\boldsymbol{b})\}, \\ R_1 + R_2 &< I(X_1,X_2;Y). \end{aligned}$$

Error Analysis: "Multiple-Access Decoding" Bound

$$\mathsf{P}\{\mathcal{E}_3 \cap \mathcal{E}_1^c\} \le \mathsf{P}\Big\{(U_1^n(m_1,l_1), U_2^n(m_2,l_2), Y^n) \in \mathcal{T}_\epsilon^{(n)},\ \mathcal{E}_1^c \text{ for some } (m_1,l_1,m_2,l_2) \ne (M_1,L_1,M_2,L_2)\Big\}$$

- Can show that $\lim_{n\to\infty} \mathsf{P}\{\mathcal{E}_3 \cap \mathcal{E}_1^c\} = 0$ if

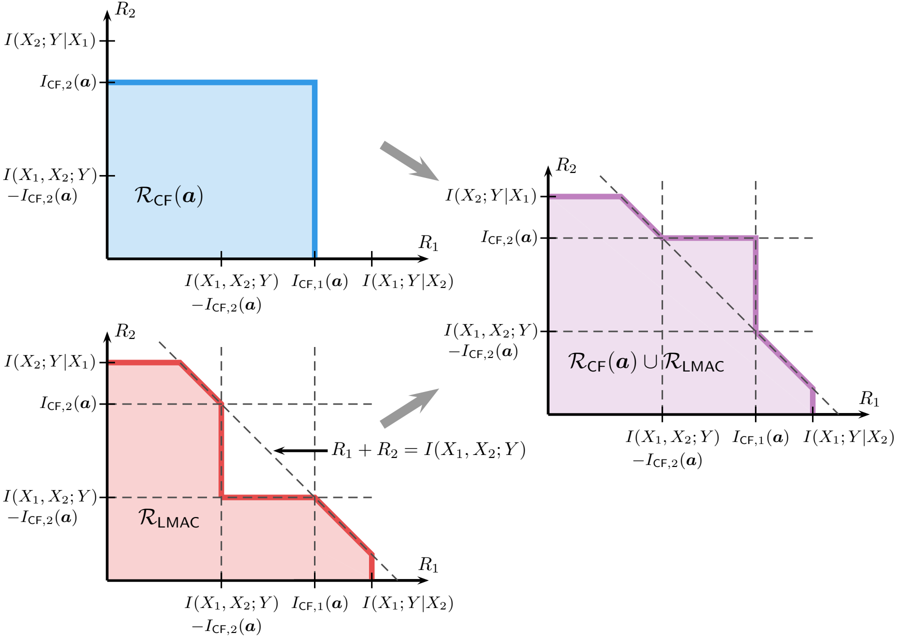
$$\begin{aligned}
R_1 &< \max_{\boldsymbol{b}\in\mathbb{A}^2\setminus\{\mathbf{0}\}} \min\{I_{\mathsf{CF},1}(\boldsymbol{b}), I(X_1,X_2;Y) - I_{\mathsf{CF},2}(\boldsymbol{b})\},\\
R_2 &< I(X_2;Y|X_1),\\
R_1 + R_2 &< I(X_1,X_2;Y)
\end{aligned}$$

OR

$$\begin{aligned}
R_1 &< I(X_1;Y|X_2),\\
R_2 &< \max_{\boldsymbol{b}\in\mathbb{A}^2\setminus\{\mathbf{0}\}} \min\{I_{\mathsf{CF},2}(\boldsymbol{b}), I(X_1,X_2;Y) - I_{\mathsf{CF},1}(\boldsymbol{b})\},\\
R_1 + R_2 &< I(X_1,X_2;Y).
\end{aligned}$$

- The $I_{\mathsf{CF},2}(\boldsymbol{b})$ term plays a key role in handling the dependencies between competing pairs of linear codewords.

Rate Region



R_2
$I(X_2;Y|X_1)$
$I_{\mathsf{CF},2}(\boldsymbol{a})$
$I(X_1,X_2;Y)$
$-I_{\mathsf{CF},2}(\boldsymbol{a})$
$\mathcal{R}_{\mathsf{CF}}(\boldsymbol{a})$
R_1
$I(X_1,X_2;Y)$
$-I_{\mathsf{CF},2}(\boldsymbol{a})$
$I_{\mathsf{CF},1}(\boldsymbol{a})$
$I(X_1;Y|X_2)$
R_2
$I(X_2;Y|X_1)$
$I_{\mathsf{CF},2}(\boldsymbol{a})$
$I(X_1,X_2;Y)$
$-I_{\mathsf{CF},2}(\boldsymbol{a})$
$\mathcal{R}_{\mathsf{LMAC}}$
$R_1 + R_2 = I(X_1,X_2;Y)$
R_1
$I(X_1,X_2;Y)$
$-I_{\mathsf{CF},2}(\boldsymbol{a})$
$I_{\mathsf{CF},1}(\boldsymbol{a})$
$I(X_1;Y|X_2)$
R_2
$I(X_2;Y|X_1)$
$I_{\mathsf{CF},2}(\boldsymbol{a})$
$I(X_1,X_2;Y)$
$-I_{\mathsf{CF},2}(\boldsymbol{a})$
$\mathcal{R}_{\mathsf{CF}}(\boldsymbol{a}) \cup \mathcal{R}_{\mathsf{LMAC}}$
R_1
$I(X_1,X_2;Y)$
$-I_{\mathsf{CF},2}(\boldsymbol{a})$
$I_{\mathsf{CF},1}(\boldsymbol{a})$
$I(X_1;Y|X_2)$

Concluding Remarks

- First steps towards bringing algebraic network information theory back into the realm of joint typicality.
- Joint decoding rate region for compute-and-forward that outperforms parallel and successive decoding.